Communications Canada
Canadian Workplace Automation
Research Centre
Organizational Research Directorate

THE COGNITIVE ERGONOMICS OF COMPUTER SYSTEMS: STATUS OF RESEARCH AND RECOMMENDED AGENDA

Luc Giroux, PhD
and
Serge Larochelle, PhD

University of Montreal

July 1987
Laval

CC-CWARC-DLR-85/6-019

This report is one of a series of research reports done by or for the Organizational Research Directorate, Canadian Workplace Automation Research Centre, Communications Canada. The views expressed in this report are those of the authors. This report was originally written in French, and was translated into English by the Translation Bureau, Department of the Secretary of State of Canada.

Copyright Minister of Supply and Services Canada 1988

Cat. no. Co28-1/7-1988E

ISBN 0-662-16250-1
(Original version: 0-662-94891-2, CCRIT, Laval)

La version française du rapport est disponible

Sommaires français et anglais aussi disponibles

Summary available in French and in English

We would like to express our thanks to Cynthia Ingenito, a student in the Master of Communications Sciences program at the University of Montreal, whose assistance throughout the project has been invaluable. Our thanks are also due to Amyot Bachand and Jeanette Bellerose of the Canadian Workplace Automation Research Centre for their careful reading of an earlier version of this report and their helpful suggestions.

Table of contents

Page

Summary 1

Introduction 7

Chapter 1 : The development of the field 10

1.1 From ergonomics to cognitive ergonomics 11
1.2 The cognitive ergonomics research community 16
1.3 The major schools 23

Chapter 2 : Towards an optimal user-computer system 32

2.1 The concept of the interface 35
2.2 The user-computer system 37
2.3 The criteria of optimal interaction 43

Chapter 3 : The system 59

3.1 The internal structure of computers 61
3.2 The physical interface 65
3.3 The cognitive interface 74

Chapter 4 : The user 85

4.1 The typical user 88
4.2 Types of user 91
4.3 The individual user 102

Chapter 5 : The task 104

5.1 Word processing 107
5.2 Databases 112
5.3 Electronic mail 121

Chapter 6 : Recommended research agenda 126

6.1 Direct manipulation interfaces 127
6.2 Mental models 130
6.3 Intelligent interfaces 133
6.4 The organizational interface 135

List of periodicals consulted 138

Bibliography 139

List of figures

Page

Figure 1 Trends in publications in cognitive ergonomics 22

Figure 2 Model of the user-computer system 38

Figure 3 Eason's "usability" model 56

Figure 4 Classification of query languages 118

Figure 5 Classification of ergonomic criteria 120

SUMMARY

Cognitive ergonomics of information processing systems is a new field. This report outlines the current status of research, and describes the principal ergonomic problems posed by new information processing technology, as well as the methods used to solve them and the results obtained. The authors also suggest new research directions likely to further knowledge in this field. The report contains six chapters. The first two describe cognitive ergonomics in relation to other fields of scientific research, and give a general outline of the problems involved. Chapters 3, 4 and 5 in turn describe the three major aspects of person-computer interaction: the information processing system, the human user, and the task. The sixth chapter contains four recommendations for further research.

Chapter 1: The development of the field

The first section of this chapter relates cognitive ergonomics to traditional ergonomics, and indicates how the advent of information processing technology has altered our concepts of tools and machines. In the past, traditional tools and machines were designed to operate on physical objects; the computer, by contrast, is essentially designed to manipulate information. This new tool has led us to redefine ergonomics as a discipline. The traditional goal of adapting the work to the person and the machine to the operator remains unaltered, but adaptation can no longer be limited to the user's perceptual and motor skills alone; it must also include his cognitive functions. Knowledge derived from disciplines such as cognitive psychology and artificial intelligence therefore become not only relevant, but also necessary in resolving ergonomic problems posed by the person-computer relationship. The next section of the chapter describes the creation of the first research centres in the field, as well as the founding

of learned societies and the appearance of specialized journals. This section also contains a list of the major research groups working on person-computer interaction. In light of the large number of disciplines interested in ergonomics research, it is hardly surprising to find several different methodological approaches, which the third and last section of the chapter divides into four broad schools of thought. The guideline school, as its name indicates, aims at establishing basic principles to guide engineers in designing systems better adapted to users. The experimental school is characterized by its use of classic scientific methodology. This approach can be directed toward a variety of goals, such as validating fundamental theories on person-computer interaction, or carrying out specific tests on systems either already in existence or currently being developed.

The calculational school focuses on developing models that will predict, in numerical terms, the consequences on user performance of variations in system design. This reduces the need for empirical testing, which often costs a great deal of time and money. Lastly, the intelligent interface school is working toward building systems that are inherently capable of adapting themselves to the user.

Chapter 2: Toward an optimal user-computer system

Although most researchers in cognitive ergonomics are trying to facilitate person-computer interaction, few can agree on how to do so. Chapter 2 attempts to define the general concept of a user-computer interface. The first section defines the interface in several ways, and shows that an overly restrictive definition, limited to the points of physical contact between user and machine, allows little scope for improving the person-computer relationship. The second section describes a general model of what we may call a user-computer system. The model has three major components: the information processing system, the user, and the task which motivates interaction. The information processing system itself has several components: the interface software, the interface hardware, and the internal hardware of the machine. For users, we must distinguish between basic cognitive abilities common to all, and differences which may exist between individuals or groups. Ideally, we should also take into account the variations that may occur in a single user at different stages of familiarization with the system, or even during a single interaction session. We should also note that the computer can perform increasingly varied tasks, which differ not only in their nature but also in the complexity of the modes of interaction. These modes are called operating languages.

The third section of the chapter lists ten criteria proposed to evaluate person-computer interaction. Some of the criteria are related to user performance: speed of executing a task, error rate, and ease with which errors can be corrected. Others are related to learning: facility with which system-related knowledge can be acquired, retained, and applied to other systems. The remaining criteria are: power and flexibility of the system, acceptability, and the degree to which it is used. Unfortunately, it is almost always impossible to satisfy all ten criteria, and often difficult to choose which to discard at any given time.

Chapter 3: The system

This chapter examines the information processing system itself. Clearly, cognitive ergonomics is mainly concerned with the interface software of any information processing system. Limitations on interface software are ultimately imposed both by the interface hardware and the computational power of the machine. Thus, the first section of the chapter briefly describes some aspects of internal computer structure. The second describes interface hardware, including principal input modes, such as keyboards and various cursor-positioning systems, and principal output modes, such as printers and screens. Some of the research on multimodal interfaces is also cited. These sections reveal the enormous technological resources available to software creators. As a result, it seems that limitations on existing interface software stem largely from gaps in our theoretical knowledge of information processing in general, and within the human brain in particular. The third section describes the cognitive interface and reviews studies illustrating the problems that users have with learning and using the types of artificial syntax and vocabulary required to interact with the computer. These problems would be partially solved if users could interact with the computer verbally and use natural language. As yet, however, there is no general procedure for recognizing speech or understanding language automatically. In addition, this section briefly covers a relatively new type of interface: concepts are represented by images, commonly called icons, which are then manipulated directly. Although such interfaces do seem easier to use, we still know little about the way in which space and images are represented in the user's mind.

Chapter 4: The user

The first section begins with a model of a typical user. Although based on 20 years of research in cognitive psychology, the model encompasses only basic cognitive skills, which all users should therefore have. The second section discusses the differences between users that have been identified and studied. One line of research is derived from classic differential psychology, and examines differences in aptitudes, skills and styles among various users of information processing systems. A second line is derived from ergonomics and classifies users by more pragmatic criteria, such as the nature of their work, motivation and level of expertise. The latter two criteria illustrate one of the problems inherent in such classifications: a single user can change category as his/her level of expertise, or even daily motivation, changes. The only way to take these variations into account is by creating more flexible interfaces which can adapt to the requirements of the user. The last section describes several different approaches to the problems of interface in artificial intelligence.

Chapter 5: The task

After a look at tasks in general, this chapter focuses on the three major tasks in office automation: word processing, creation and retrieval of data in data banks, and electronic mail. The first section discusses text editing, which undoubtedly has received the most attention in cognitive ergonomics. Studies presenting widely differing viewpoints are also discussed. The next section examines data banks, and reveals that whereas most problems encountered in word processing are rooted in the syntax of a system, data banks pose mainly semantic problems. Various ways of organizing information in data banks are described, as well as various modes of access. The last section deals with electronic mail, and considers at some length the problem of electronic junk mail: how to discriminate between relevant and irrelevant messages. This may seem of little importance, but in a world where communication is electronic, lack of selectivity in distributing messages can cost a great deal of time and money. However, if selectivity were too strict, important messages might be prevented from reaching their proper destinations. The section proposes several solutions.

Chapter 6: Lines of research

The last chapter is deliberately more speculative than the first five. Here the authors describe the lines of research they feel will contribute most toward the development of optimal interaction. Their first recommendation is for research on direct manipulation interfaces, since this is a new field which needs to be developed and more widely applied. However, very little is known about the cognitive advantages of such interfaces or the way they are represented in the user's mind.

Their second recommendation is for the study of mental models developed by users. We must gain a better understanding of how users perceive the system with which they interact. Several studies reveal that users generally have inadequate mental models, which result in non-optimal use of the system and paralyze users when they make an error.

The third recommendation concerns the development of intelligent interfaces, which derive from research in artificial intelligence. It is important for systems to have a better inherent "understanding" of the user and task. The authors therefore recommend that artificial intelligence research be pursued and applied to interface development even if there are no immediate benefits.

The fourth and last recommendation is for the study of what may be called the "organizational interface," the way in which people organize their work individually and collectively. There are various indications that the quality of interaction depends not only on the system, user and task, but also on the organizational context in which interaction takes place; for example, secretaries are often subject to constraints which limit mastery of their machines, such as management being unwilling to allow the time necessary for adequate training. It is also essential to analyse how the use of information processing technology fits into the general context of the user's tasks, so that its impact on the overall productivity of both user and organization can be assessed.

INTRODUCTION

There has been a great deal of discussion in recent years of the rapid strides in computer technologies, to the point where the term has become almost a cliché.

In fact, however, the cliché has become a reality: computer technology has undergone an absolutely astounding evolution in recent years, perhaps the most dramatic in all the annals of human history. According to Birnbaum (1982) the ratio of hardware performance to cost increased by a factor of one million between 1955 and 1985. By comparison, it is estimated that if the aeronautics industry had undergone a similar evolution over the past 25 years, a Boeing 767 would now cost $500 and could fly around the world on twenty litres of fuel (Toong and Gupta, 1982).

There have been two particularly significant consequences of miniaturization and the ever-increasing power of hardware: the proliferation of applications and the diversification of users.

The sole function of the first computers, such as ENIAC, developed in 1946, was to perform ballistic calculations for military purposes. At that time, it was thought that computers would be used strictly for complex mathematical calculations, so that in the eyes of many people the **Information Age** had ended before it had really begun: Bowden (1970), quoted in Nickerson (1986) recounts how in 1950 he consulted an eminent British expert to evaluate the potential viability of a company specializing in the manufacture of computers for commercial purposes. He was told that the company would be doomed to bankruptcy, since the three computers then in existence would be more than sufficient to perform all the calculations that England would ever need. Other contemporary observers estimated that a dozen machines would satisfy the entire U.S. market (Diebold, 1969, quoted in Nickerson, 1986).

These anecdotes make us smile today, in an age rife with speculation on the role of computers in virtually every sphere of human activity. Our purpose here is not to extrapolate about the innumerable applications of computers and their impact, since many others have trodden this path, including Abelson (1982), the Science Council of Canada (1982), Cordell (1985), Evans (1980), Godfrey, Madden, Parkhill and Ouimet (1980), Naisbitt (1984) and Toffler (1980). Regardless of the scenario envisaged, however, it is clear that computer-related technologies are in the throes of rapid expansion and will remain so for the foreseeable future.

The second consequence of hardware development is a corollary to the first: by diversifying its applications, computer technology has multiplied the number of its potential users. If at one time it was possible to characterize, however generally, the typical computer user as an expert in programming willing to accept all the inherent frustrations to obtain the desired result, contemporary users have become indescribable by virtue of their variety: level of motivation, level of technical expertise and frequency of use are only some of the variables involved, quite apart from the transformation that individual users undergo as a result of practice. One thing is clear, however: today's user is far less tolerant and demands a system which is much easier to master and to use. In turn, these major changes in the potential of computers have had a marked impact on the economy of the market. Severe competition has resulted in the industry shifting its orientation towards marketing, thereby forcing designers to respond effectively to their customers' requirements, rather than simply offering more powerful machines. According to Burch (1984), the industry is now more "market driven" than "technology driven." There has been discussion for some time of trends in the relative cost of hardware and software, with the latter occupying an ever-more important position. The time has now come to consider the human costs of computer technology: in a wide range of applications, and especially in an office context, the cost of investment in computers is dropping to the point where it is negligeable in comparison to the salaries of the employees who use them. The human element is tending to become the key factor in what can conveniently be termed the person-machine system. We must therefore analyse the resources and the limits of the individual and develop the technological element so that it maximizes the potential of its human partner. This is why the study of the "human factors" occupies an increasingly dominant position in the planning, design and marketing of modern computer systems.

This study is intended as a critical presentation of what has come to be called the "cognitive ergonomics" of computer systems, which represents a major re-orientation in contemporary research on ergonomics. Our first stage will be an attempt to define the field of cognitive ergonomics by locating it in its theoretical and practical context. After a historical overview and definition of the basic parameters of this applied science, we will present a model of the person-computer system which will provide an analytical framework for the second part of the study, in which we will review the present state of our knowledge and the major goals of current research. We shall conclude with a critical review of the problems involved in cognitive ergonomics and the formulation of a number of recommendations with regard to the research agenda.

Chapter 1

The development of the field

1.1 From ergonomics to cognitive ergonomics

Ergonomics

Ergonomics is a very recent discipline: the term itself (taken from the Greek **ergon** (work) and **nomos** (natural laws)) was "officially" adopted in 1949 when the Ergonomic Research Society was founded in England. The terms "human engineering" or "human factors" were preferred in the United States.

Ergonomics as a discipline grew under the pressure of the technological developments of the first half of the twentieth century, which significantly increased not only the power but also the complexity of the machines available: whether one thinks of the number of dials and levers in an aircraft cockpit, or, on a simpler level, the act of driving a car, which often requires the simultaneous manipulation of several controls. Clearly, the operator's physiological and perceptive capacities had to be taken into account for such machines to be usable.

Thus ergonomics was born, a discipline which might be described as "the collection of knowledge of the functioning of man in operation, in order to apply them to the design of tasks, tools, machines and systems of production" (Laville, 1976), or as the science of "adapting work to man, in other words, adapting the workplace, the tools, machines, schedules and the environment to his requirements" (Grandjean, 1969).

Regardless of the definition adopted, the basic premise remains the same: ergonomics is primarily an applied human science, the purpose of which is to change technology to adapt it to human beings, rather than the reverse. The nuance is crucial: ergonomics is not oriented towards personnel training problems. The idea is not to train a machine operator so that he can become more efficient (even though training is of course not excluded), but rather to study human capacities and build a machine which is a function of these capacities, so that the performance of the human-machine system is safe, efficient and reliable.

The efforts of three groups of disciplines have converged in ergonomics: the **biological sciences** (anatomy, physiology, medicine) supply the knowledge about the human being's physical capacities and limits; strength, resistance to effort and fatigue, etc. **Experimental psychology** and **psychophysiology** study the constraints imposed by the nervous system, perceptive and motor capacities as well as learning capacities. **Physics** and **industrial engineering** supply the tools to measure and modify technology and the environment in which the task is performed (Osborne, 1982). Today, the field of "classical" ergonomics has acquired an impressive panoply of methods and tools: load analysis, anthropometric tables, computer-generated reach curves and so on (Kantowitz and Sorkin, 1983; McCormick and Sanders, 1982; Roebuck, Kroemer and Thompson, 1975).

However, like to the stimulus produced by the events of World War II, technology has again taken vast strides: in the era of microcomputers, traditional ergonomics are no longer sufficient. This is why a discipline has developed in recent years (which may be considered a sub-discipline of ergonomics or not, depending on individual preference), devoted to the same goals, but centred on the specific characteristics of computer technologies and drawing primarily on **cognitive psychology**, **computational linguistics** and, increasingly, **artificial intelligence**. This discipline is known by the name **cognitive ergonomics**, and is also called "user science" (Vallée, 1976; see also Card, Moran and Newell, 1983), "software psychology," Shneiderman, 1980a), "user psychology" (Moran, 1981a) and "cognitive engineering" (Norman, 1980; Hollnagel and Woods, 1983).

Cognitive ergonomics

As one might suspect, the boundary between traditional and cognitive ergonomics is not clearly defined. In fact, it is less a question of the nature of the discipline than of emphasis, as will become clearer from the contrast between classical industrial technologies and computer technology.

The development of microcomputer technology has radically transformed the type of machine which human beings will henceforth be called upon to operate. Whereas the function of an industrial machine is generally to manipulate **objects** in space or to act on physical forces (the automobile and the mechanical crane are typical examples of this), the computer manipulates **information**. The focus has therefore shifted from physical objects to chains of symbols, altering both the complexity and the nature of the interchange between the operator and the machine (Giroux, 1983).

The first thing to remember is that industrial machines are generally single-function: whether the machine in question is a sewing machine or an automobile, variations in commands remain centred on the performance of a single task, which is the raison d'être of the industrial technical object. Computing machines, on the other hand, are, by their very nature, Turing machines (Turing, 1950) which can be dedicated to a theoretically infinite number of applications. In concrete terms, it is easy to find examples of simple machines which have become multifunctional with the advent of the microprocessor. As Card (1984) has said, it will soon be necessary to equip a simple thermostat with an alphanumerical keyboard in order to program all the functions of which it is capable. Ergonomics is therefore faced with analysing tasks which are more varied as well as more complex and which have to be integrated into a single working environment.

The computer has also changed the nature of the interface between humans/people and technology. The relationship has taken on the form of an interaction which is close to a dialogue in real time and is no longer conducted according to the same principles as before. In the interface with an industrial machine, the perceptual and psychomotor elements are omnipresent and the controls are analog: the manipulation of levers, dials and pedals, perception of movement and shapes, etc. Computers have however from the outset imposed a "digitalized" interaction (although we will see later on how there is a trend towards a return to the analog mode) and the cognitive element has become central: the tasks themselves are highly cognitive by nature (programming, word processing, data retrieval, etc.) and the interaction is henceforth symbolic. The aim is no longer the operation of a machine which obeys, but a dialogue with a system which displays several of the characteristics of a true interlocutor (Nickerson, 1977). Interaction with a computer is primarily intellectual, and the focus therefore had to be shifted towards **the study of the mental processes** at work. For example, perceptual problems related to the reading of a menu on a screen will not be entirely eliminated, but the primary focus of attention will be on the attempt to understand the cognitive processes surrounding the reading: what is the role of the information or the command sought in relation to the user's strategy? Is consulting the menu a recognition or classification task for the user? Will the way in which he scans the menu be influenced by the nature of his task? If so, does the way in which the menu headings are organized encourage this method? (Giroux, 1986; Giroux and Belleau, 1986; Giroux, Bergeron and Lamarche, unpublished).

Cognitive ergonomics and cognitive science

Many researchers in **cognitive science** have quickly and almost naturally become experts in the interaction that takes place between the individual and the computer. Cognitive science, which sits at the junction of cognitive psychology, computational linguistics and the branch of computer science known as artificial intelligence, embraces the foundations of human cognition, in which the intellectual mechanisms are construed as **a system of processing symbolic information**. From this perspective, behavior is seen as being directed by an internal model of the environment constructed by the subject, who applies the model to plan his actions, make decisions, formulate the messages to be transmitted and interpret the messages received (Hollnagel and Woods, 1983; Johnson-Laird, 1980).

This almost spontaneous interest on the part of a number of researchers in cognition in the direction of the psychological dimension of the use of computers is understandable enough. Computers have long been a routine working tool in cognitive psychology laboratories, where they are used to monitor experiments, many of which deal with response time (Lachman, Lachman and Butterfield, 1979), a favorite dependent variable. They are also used to perform computer simulations which are regarded as a particularly effective method of empirically validating theories. Furthermore, the very nature of the tasks done on computers, which are performed within a limited and well-defined set of rules, makes the human-computer interaction a favored area to apply theories about human information processing since such theories attempt to describe precisely behavior from the point of view of "effective procedures" or programs (Newell and Simon, 1972). In return, human-computer interaction provides empirical data which have never before been observed and has the additional advantage of using a human interlocutor and a partner all of whose parameters are known.

The new "cognitive ergonomists" have therefore focused on the three main elements of the human-computer system: the user, the computer system and the task to be performed. As far as the user is concerned, the focus is on short-term memory, the representation in long-term memory, attention, learning, problem-solving, planning and language. These were all already preferred fields of study. With regard to the computer system, the requirement for sufficiency set by information processing theoreticians (in other words, achieving a degree of precision sufficient to simulate behavior or reproduce it symbolically) has resulted in the current tendency to formalize models in the form of programs. Since the computer system, from this standpoint, has the features of a symbolic information processing system, it can be examined in the same way: memory structure, heuristics of processing, study of the features of artificial languages, etc. Cognitive science also supplies new methods of describing and analysing tasks, not only in terms of performance, but also in terms of the process of problem-solving or routine cognitive exercises (Ericsson and Simon, 1985).

Research in cognitive ergonomics is a booming field and the number of research centres around the world is growing.

1.2 The cognitive ergonomics research community

In view of the newness of the field and its rapid growth (the pioneers of cognitive ergonomics are, almost without exception, still actively engaged in research), it is difficult to paint a clear picture of the research community as it exists at present, let alone a complete one. We will therefore refer to a brief overview by Burch (1984) and one by Shackel (1985), supplemented by our personal knowledge of the field.

The first centres

The earliest works on the ergonomics of computer systems go back to the early 1960s (see among others English and Engelbart, 1967; Licklider, 1960, in Nickerson, 1986). The first congress on the question of human-computer interaction seems to be the "IBM Scientific Computing Symposium on Man-Machine Communication," held in Yorktown Heights in 1965. It was not until the beginning of the 1970s, however, that a number of research groups emerged whose orientation was focused more specifically on the cognitive aspects of the ergonomics of computer systems.

The year 1970 marked a decisive turning point in the field of cognitive ergonomics, since it saw the establishment of two important centres of research. At Loughborough University in England, B. Shackel founded the Research Group on Human Sciences and Advanced Technology (HUSAT), which became a training ground for many of those currently engaged in research and the incubator of a number of major works. At the same time, Xerox Corporation founded the Palo Alto Research Center (PARC), which would attract researchers with an international reputation in several fields, such as cognitive psychology, artificial intelligence and system design. The centre was to acquire a worldwide reputation, due among other things to the development of "direct manipulation" interfaces. PARC became the home of the "Applied Information Processing-Psychology Project," established in 1974, which has produced some of today's leading theoreticians in cognitive ergonomics.

The learned societies

Until the early 1980s, researchers in cognitive ergonomics were known primarily within the major learned societies in ergonomics, the Human Factors Society in the United States and the Ergonomic Society in England. In recent years, however, researchers have combined to form more autonomous societies: the Association for Computing Machinery (ACM), which is without a doubt the oldest and most important computer society in America, if not the world, founded the Special Interest Group in Computer-Human Interaction (SIGCHI), which has held an annual Conference every year since 1981. More than 1,500 participants attended the 1986 CHI'86 congress.

Development in Europe has been slower. Although several important conferences have been held since 1982 (European Conference on Cognitive Ergonomics (ECCE) in Amsterdam in 1982, in Gmünden in 1984 and Paris in 1986; INTERACT'84 in London; MACINTER-1 in Berlin in 1984), no society was formed devoted exclusively to cognitive ergonomics. On September 18, 1986, the participants at ECCE in Paris decided to form the European Association for Cognitive Ergonomics (EACE), which was officially founded at INTERACT'87 in Stuttgart.

An initial attempt to form an intercontinental association occurred in Hawaii in 1984 during the first Japanese/American Joint Conference on Human-Computer Interaction (HCI). Since there were more international participants than expected, the conference became the International Conference on Human-Computer Interaction (Hawaii, August 1987).

Several other conferences have been held on almost every continent, and a number of learned societies are thought to be in the process of gestation. Although it will take a few years before the situation stabilizes, the scope of this activity is eloquent testimony to the rapid growth of the field.

Publications

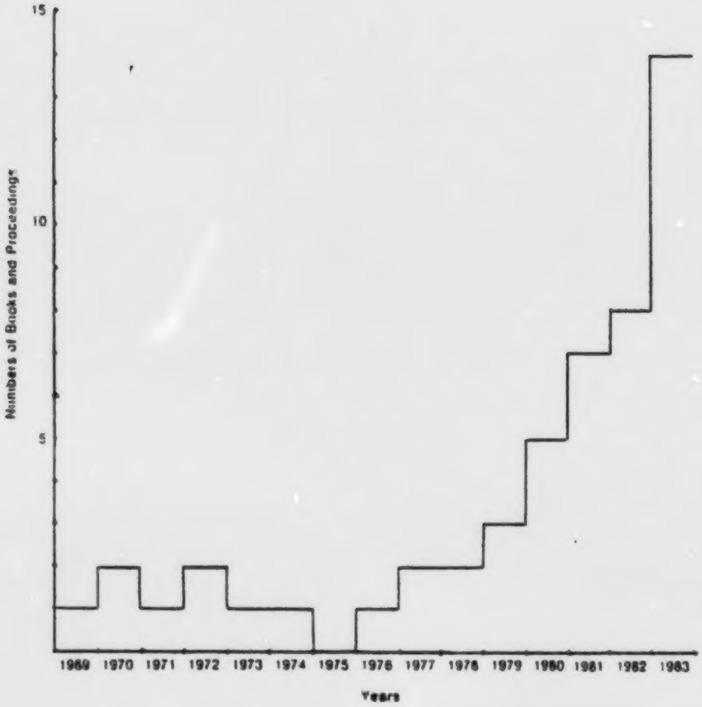
The growth of cognitive ergonomics has been accompanied by a marked change in the content of the main ergonomics publications (Le Travail Humain, Applied Ergonomics, Ergonomics and Human Factors) and some of the computer periodicals, such as the International Journal of Man-Machine Studies. As in the case of the learned societies, the publications tend to dissociate themselves from traditional ergonomics, as is reflected in the appearance of such periodicals as Behaviour and Information Technology (BIT) and Human-Computer Interaction (HCI).

Since the appearance of Martin's book (1973) Design of Man-Computer Dialogue, there has been an explosion of publications in the form of books and conference proceedings. Several recent volumes have titles such as The Psychology of Human-Computer Interaction (Card et al., 1983), Human Factors and Interactive Computer Systems (Vassiliou, 1984), Human Factors in Computer Systems (Thomas and Schneider, 1984), Software Psychology (Shneiderman, 1980a), Advances in Human-Computer Interaction (Hartson, 1985), User Centered System Design (Norman and Draper, 1986) and Using Computers (Nickerson, 1986). Figure 1, which is taken from Shackel (1984), illustrates the rapid growth in the number of books and conference proceedings on the ergonomics of computer systems. Although not all of them are devoted exclusively to cognitive problems, these nevertheless figure prominently throughout.

Existing centres

The field of cognitive ergonomics has developed far more rapidly in the United States than in Europe, primarily because of industry support. Whereas most research in Europe is carried on in university laboratories and government agencies, the major American computer firms such as Xerox, DEC and IBM soon seized the initiative, to the extent that "interface design" has become a not-inconsiderable phase of new product development. There has also been a massive injection of military funds into private-sector research.

Figure 1 - Trends in publications in cognitive ergonomics (Shackel, 1985)



The reader will find below a short list of the main research establishments that we have identified in Europe and North America. The list makes no claim to be exhaustive: only some of the best-known private-sector and university centres have been included. In several cases, the name of one of the active researchers appears in brackets by way of reference.

Canada

- Behavioural Research Group, Department of Communications (T. Whalen)
- Carleton University (J. Tombaugh)
- Computer Systems Research Institute, University of Toronto (W. Buxton)
- University of Calgary (B. Gaines)
- University of Victoria (E. Lee)

United States

Universities

- Carnegie-Mellon University (A. Newell)
- George Washington University (J. Foley)
- Johns Hopkins University (J. Chapanis)
- Massachussetts Institute of Technology (R. Bolt and N. Negroponte)
- University of California at San Diego (D. Norman)
- University of Maryland (B. Shneiderman)
- Virginia Institute of Technology (R. Williges)
- Yale University (E. Soloway)

Corporations

- AT&T Bell Labs
- Bell Communication Research (T. Landauer)
- Bolt, Beranek & Newman Labs Inc. (R. Nickerson)
- Digital Equipment Corporation (J. Whiteside)

- Hewlett-Packard Labs (J. Rosenberg)
- Microelectronics and Computer Technology Corporation (MCC) (B. Curtis)
- User Interface Institute, IBM Corp. (J. Gould)
- Xerox Corporation (J.S. Brown)

France

- Institut National de la Recherche en Informatique et en Automatique (INRIA) (national research institute on informatics and automation), Rocquencourt, Sophia-Antipolis and Rennes (P. Falzon)
- Laboratoire de Psychologie du travail (work psychology laboratory) of the École Pratique des Hautes Études (EPHE), Paris (J.M. Hoc)

Great Britain

- Human Computer Interface Research Unit, Leicester Polytechnic, Leicester
- HUSAT, University of Technology, Loughborough (B. Shackel)
- MRC Applied Psychology Unit, Cambridge, (T.R.G. Green)
- University of Lancaster (S. Payne)

Federal Republic of Germany

- Fraunnhofer-Institut für Arbeitswirtschaft und Organisation (Fraunnhofer Institute for ergonomics and organization), Stuttgart (J.E. Ziegler)
- IBM Wissenschaftliches Zentrum (science centre), Software Ergonomics Group, Heidelberg (G. Rohr)
- Technische Hochschule (technical university), Aachen (N.A. Streitz)
- Humboldt University, Berlin (F. Klix)

Switzerland

- Institut für Informatik, Zürich

Sweden

- University of Göteborg (C.M. Allwood)
- University of Stockholm (Y. Waern)

Denmark

- Riso National Laboratory (E. Hollnagel)

Netherlands

- Free University, Amsterdam (G.C. Van der Veer)

Italy

- Instituto di Psicologica del CNR, Rome, (S. Bagnara)

1.3 The major schools

Cognitive ergonomics is a science applied to the optimization of human-computer interaction; the next question is therefore **when** and **how** it approaches the object of its investigation.

There are three **moments** at which intervention is possible: before, during and after the user interface has been established. Prior to the test phase, in other words, at the point at which the system is designed "on paper," the designer will be guided by principles and tools provided by ergonomics. During the prototype testing phase, it will provide methods for testing design alternatives or to pinpoint problems encountered by users. Once the design has been finalized, ergonomics should make available to decision-makers tools with the capacity to measure the actual efficiency of the product and to conduct a comparative evaluation of the systems.

Clearly, cognitive ergonomics can make its contribution most economically in the initial design phase, since every gain at this stage shortens the design-testing-modification cycle commensurately. Furthermore, the procedure for developing and testing the interface is still poorly defined, and a great deal of current work focuses on this problem of **design methodology**, which we will not go into here.

As far as the **type** of intervention is concerned, there are four approaches in the field: the **guidelines** school, the **experimental** school, the **computational** school and what we call the **"intelligent interface"** school. The distinction we use here is not a normal typology in the field, although it is based partly on Moran's analysis (1981a). In any event, it seems to us to offer a relevant grid for understanding and reducing the apparent disparity between current research projects.

The guidelines school

In their survey of system designers, Smith and Mosier (1984) show that they estimate that approximately 35 percent of a system's software is devoted to the user interface. Clearly, the design of this part of the software is one of their key functions, both from the point of view of the energy and cost of development and that of the effect on the product's efficiency. The "guidelines" approach provides a direct response to this concern on the part of designers: to have available principles, guidelines, rules, even algorithms or standards for the design of the interface. We would stress the importance of the distinction between standards and guidelines: standards are generally formal rules imposed by legislation or by order of the contractor, whereas guidelines are merely more or less general principles, on which a certain consensus exists, but which each individual is free to adapt to his own specific needs. For the moment, the standards exist primarily in the area of hardware (such as the number of lines on the screen, keyboard configuration, etc.) and our knowledge of the interface is still too vague to enable the guidelines that have been formulated to become standards (Smith, 1986).

In light of the above, the works which generate lists of guidelines to follow are without a doubt the most influential at present, since they respond to the most pressing needs of their users. This is the reason for the proliferation of such lists in the literature (see among others the overview compiled by Maguire, 1982a, and the bibliography given in Nickerson, 1986, p. 225), even though many authors consider it premature and even harmful to refer too easily to what they regard as recipes with no rigorous foundation (Norman, 1983; Payne, 1985).

The guidelines run from extremely general advice ("take the user into account," "be consistent," etc.; Heckel, 1984) to much more precise recommendations on display formats, the formulation of error messages or the chunking of strings of characters (Harris, 1986), by way of intermediate stages of generality (Gaines and Shaw, 1983). Some are formulated in light of system evaluations, while others are derived from experimental results;

the majority, however, are produced by experienced researchers who base their recommendations on their intuition and knowledge.

The use of such guidelines has major advantages as well as obvious limitations. The guidelines are often based on empirical observation of actual cases of success or failure. They constitute an accessible source of documentation and a working tool for designers who have to solve concrete problems in the short term. They are often the only feasible alternative solution during the design phase of the interface, since experimental data almost invariably arrive too late, in other words, when the crucial choices have already been made. The advocates of this school claim that, under present circumstances, the interface is an uncertain world for which no general, reliable theory exists; like it or not, the quality of design is based in large measure on the designer's judgment. Ultimately, it is better for him to be able to rely on generally accepted principles than on his intuition alone (Smith, 1986).

The use of guidelines is nevertheless not unproblematic. On a more theoretical level, the principles distract us from fundamental questions: they are rarely validated and some of them have been proven simply wrong in concrete situations. They are often formulated on the basis of a few observations and then generalized for all cases, without taking into account the interaction with a host of other factors, such as the type of user and the nature of the application. This abusive generalization sometimes results in contradictory recommendations, as Maguire (1982a) points out, and ignores the question of the trade-offs required with regard to the relative priority of the dimensions to be optimized (Norman, 1983).

In practical terms, the guidelines are far from entirely satisfactory for those for whom they are intended, even if they are considered useful. Mosier and Smith (1986) interviewed 130 people involved in design, who had been given a list of guidelines one year before. On average, the respondents, stated that they found the guidelines useful, had used them and were prepared to do so again, although they pointed out several problem areas: 60 percent of the guidelines offered had not been applied. The reasons given for this varied widely: an incomprehensible guideline, which appeared to be inapplicable, irrelevant or false; a guideline which seemed too general or too specific, guidelines which seemed difficult to reconcile, etc. The authors conclude that, although the approach is useful, users require an interpretive framework to enable them to turn the guidelines into solutions that are applicable to their own specific cases.

It seems clear that the guidelines school is firmly established, and that its usefulness is incontrovertible. The value of the principles formulated will nevertheless always be limited by the frontiers of our understanding, frontiers which are above all dependent on work of a more theoretical nature.

It should nevertheless be noted that the use of guidelines does not involve a refusal to subject the choices to empirical testing.

The experimental school

The experimental school is based on classical scientific methodology, and is oriented around the testing of hypotheses and the rigorous comparison of the conditions tested. It plays a triple role: in the laboratories, it is used to validate the more fundamental models developed by the theoreticians (Rohr, 1986, for example). During the design phase, the possibilities proposed by the design team are tested (for example, Bewley, Roberts, Schroit and Verplank, 1983). Experientation is also used to compare the actual performance of systems already on the market (such as Whiteside, Jones, Levy and Wixon, 1985).

It is important to distinguish between the more theoretical experimental approach (intended to validate models) from research intended simply to compare alternative possibilities. The theoretical approach is by nature more fundamental and is oriented towards the development of general models of human-computer interaction. Although not intended for immediate application, this research is fundamental to the progress of the discipline, at least at this stage of its development. As we will see later, general theoretical models are still a rare commodity, and those involved in their development should in our view base their work on reliable empirical data.

Unfortunately, the current climate of research is not conducive to this type of attitude, particularly in the United States, where research is funded to a considerable extent by private enterprise and is focused on immediate problems. Only very large corporations and government agencies can permit themselves the luxury of longer-term investments.

The second type of experimental research is markedly more empirical and is aimed at more immediate application. The focus is on testing alternatives (command X or command Y, commands or icons, mouse or touchscreen, etc.) from the perspective of design decisions. Despite the advantage of rigorous proof, this research all too often remains atheoretical (Moran, 1981a). It is nevertheless extremely useful in the short term, since it meets the urgent requirement to move away from designs based on intuition alone. The method is expensive, however, especially in light of the lack of generality in the results, which are often applicable only in the very specific contexts in which they were obtained.

Some experts doubt the ability of experimental science to develop systems as complex and multidimensional as the user interface. As Payne stated (1985) when paraphrasing Newell's famous criticism (1973): "You can't play twenty questions with machine design and win."

Two noteworthy types of work do not always rely on the experimental method in the strict sense, but are nevertheless based on verifiable, empirical data. The first deals with the development of standardized benchmark sets of tests for comparative analysis of operating systems or applications software. Their aim, obviously, is to facilitate decisionmaking regarding the adoption of alternative systems to produce comparable evaluations. Especially noteworthy is the work of Roberts (1979; Roberts and Moran, 1982) on the evaluation of text editors, and Borenstein's critique (1985).

The other type of work is intended to develop design interface environments, in other words, software which makes it possible to simulate several virtual interfaces in order to evaluate them subsequently. Use of this procedure has the advantage of reducing considerably the process of design adjustment, since the designer has only to program each alternative in source-code (see *inter alia* the work of Henderson (1986), and Hix and Hartson (1986), as well as the statement of principle by Nakatani and Rohrlich (1983).

The computational school

The computational school is doubtless the one which is derived most directly from contemporary cognitive psychology, in other words from the paradigm of the "human processing of information." One of the precursors of this approach, Tom Moran, defines it in these terms:

> "In the calculational approach, psychological theory is cast in the form of explicit information-processing models, which spell out step by step the mental operations that the user must go through to accomplish given tasks. The models must be based on a psychologically well-based theory of the user, so that their applicability and boundary conditions, that is, the situations for which the models are appropriate to use, are clear. In order to apply such a model, a given task situation must be represented as a task (or set of tasks) from the user point of view. What the model provides is a vocabulary and framework for analysing the task into its component psychological processes. If the model is explicit enough, the system designer should be able to do this himself. By doing such an encoding into the model's framework, the designer is tailoring the psychological model to the task involved in his specific design situation. Once encoded, user behavior in this task can be calculated. Further, the designer can by calculation explore the consequences of varying the task structure (e.g. alternative interface designs). If the model produces useful information and is easy enough to use, the designer will actually use it to help him generate and evaluate alternative designs -- it will become one of his design tools."
>
> (Moran, 1981a, pp. 6-7)

The approach thus consists of initially establishing a model of the mental operations necessary to the performance of a task on the basis of a theory of cognitive functioning and a detailed examination of the subject's behavior. A model of the structure of the task can subsequently be developed from the perspective of the subject and may be of varying degrees of refinement. After establishing by approximation the time required to perform each of the "primitive actions" of the psychological model, it becomes possible to calculate, within a bracket of predictable errors, an estimate of the time required to perform the task or an unedited variation thereof. The approach is based in part on industrial engineering methods.

Card, Moran and Newell develop this approach in a major work (Card et al., 1983) in which they pay particular attention to text editing. Using their GOMS model, they succeed in predicting with considerable accuracy the sequences of actions and performance times. Unfortunately, they also demonstrate the perhaps temporary but nonetheless important limitations of their approach: despite their consummate mastery of the theories and methods of their science, the development of their model took them several years of unremitting effort. The analysis was limited to the correction of the manuscript by experts and ignores the errors made.

The calculational approach is therefore promising because it addresses the very heart of the problem, namely the extent to which its focus on the subject takes the form of problem-solving or a routine cognitive exercise. It has nevertheless proved disappointing in its concrete applications and few researchers appear to have had the courage (or the temerity) to pursue their research in this direction.

The "intelligent interfaces" school

The recent rebirth of the popularity of work on artificial intelligence has opened the door to another way of perceiving the ergonomics of interfaces: that of making them collaborators (in the true sense of the word) by endowing them with a degree of intelligence. There is some justice in this, for a number of researchers have concentrated on the problem of the user interface in expert systems; conversely, the focus has now shifted to the chances of adding an expert element to the interface.

Clearly, the creation of intelligent systems dedicated to communication with the user would solve many problems: systems capable of recognizing the user's intentions on the basis of his actions, and thus of anticipating errors, capable of accurately diagnosing errors and even -- why not? -- correcting them, systems capable of conversing in natural languages, etc. The ideal interface would possess expertise in:

1) knowledge of the user (level of expertise, personal characteristics, etc.);
2) knowledge of the task domain;
3) inferring the user's intentions and aims with regard to the problem at hand;
4) didactic principles;
5) the rules of dialogue and communication.

An interface with these capabilities would in fact solve the majority of the problems encountered in cognitive ergonomics. In fact, the question might even be asked, as Card (1984) has done, whether a user would still be needed...

"Knowledge-based interfaces" are still a very recent innovation and do not yet have a precise status in terms of research in ergonomics and artificial intelligence. Their development has nonetheless been rapid, as is reflected in recent work in the fields of oral interaction (Bolt, 1984; Schmandt and Hulteen, 1982), conversation (Reichmann, 1985), file management (Jones, 1986), help systems (Quinn and Russell, 1986; Fisher, Lemke and Schwab, 1985); tutorial and consulting systems (Chin, 1986; Nakatani, Egan, Ruedisueli, Hawley and Lewart, 1986) and plan recognition systems (Sullivan and Cohen, 1985; Desmarais, Giroux and Larochelle, 1986), as well as more general work on the design of this type of interface (Card, 1984; Mark, 1986; Rissland, 1984).

Clearly, there will continue to be an increase in research using artificial intelligence techniques for the development of intelligent interfaces. Nevertheless, with the exception of certain special programs in limited fields, the development of this type of interface can occur only when the models of the user, the field, the task, etc. have become explicit, as we discussed above. The development of interfaces or expert systems is one thing: the development of powerful theoretical models is quite another. Circumspection is called for in any discussion of the next applications of artificial intelligence. As in the 1960s, it may make promises that cannot be kept (Dreyfus, 1979; Winograd, Davis and Dreyfus, 1985; Winograd and Flores, 1986). We shall return later to research in progress in this field, for despite the above reservations it seems to us an inevitable outcome of the problems of the user-friendliness of the interface.

Chapter 2

Towards an optimal user-computer system

As we saw in the preceding chapter, cognitive ergonomics is a young albeit vigorous stripling among research fields. However, it is not yet a science with a coherent set of theories as a unifying factor. In fact, it constitutes a field whose frontiers are as yet poorly defined and whose ultimate objective cannot yet be determined with any precision, since no one knows what an optimal interaction will be. It is therefore an applied science which is pushing its own boundaries farther back as it develops and as the subject of its attention evolves.

The situation is understandable. It must be understood that cognitive ergonomics was born out of a very concrete set of problems: to develop systems and software which would be more "user-friendly" (a term to which we shall return later) and thus perform better, from the standpoint of increasing productivity and of market development. At present, a majority of those involved in the field are working within one of the major computer companies, often as system designers seeking either solutions to their problems or creative ideas likely to lead to innovations in the area of user-interfaces. The debate continues as to whether the design of user interfaces should be a science of the user or an industrial design art (Card and Norman, 1986). Although there is a consensus regarding the desirability of developing a general theory of cognitive ergonomics, there is still too little interest in the theoretical foundation of the user-computer interaction.

This is why we have hitherto refrained from giving a simple definition of cognitive ergonomics; to our knowledge, such a definition still does not exist. Cognitive ergonomics got its name because it is a science of man at work, and because the focus of its interest, the user-computer interaction, demands that cognitive factors be taken into consideration. Cognitive ergonomics is and will be whatever is thought to be needed for the development of a harmonious interaction between these two information processing systems -- humans and computers.

It is, however, possible to specify what it excludes for the present, while not presuming to know what will be the case in the future.

Cognitive ergonomics is not much concerned with the problems of traditional ergonomics, except to study their cognitive consequences. For example, the problem of keyboard configuration or the effects on the eyesight of working on a screen will be dealt with primarily for their effects on the mnemonics of function keys or, in the case of screen work, the effects on concentration and information retrieval.

One can also say that cognitive ergonomics is centred on the individual rather than on his social and professional functions, which serve only to define him as a user. It is thus little concerned with the organizational context in which the interaction occurs. As we shall see later (in the concluding chapter), it is in our view most important that this dimension be taken into account, especially in forecasting the actual use of a system in the workplace. Similarly, the most subjective aspects (technophobia, upgrading of work) will be covered primarily in terms of their effect on motivation.

What are the objectives of cognitive ergonomics? On a very general level, there is agreement with regard to the major objectives: making computer systems easy to learn, pleasant to use, and above all **appropriate to the cognitive skills** of their users -- in short, systems in which the efficiency of the interaction will be optimal. It is, however, far more difficult to operationalize these objectives. In the following sections, we shall therefore attempt to clarify certain fundamental concepts. We shall begin with an examination of the concept of the interface, which will lead us to propose a little model which we will call the "user-computer system." We shall then attempt to expound on the criteria currently used to evaluate the efficiency of this system.

2.1 The concept of the interface

When cognitive ergonomics are called upon to become involved in system design, it is generally to develop the element known as the computer's "user interface." There are, however, several ways to conceive of the interface, as the following definitions show.

> "Interface: a term used in the ergonomics of machine design, especially today in the design of computer terminals: the keyboard, the input mechanism and especially the screen and what appears on it constitute the interface between man and the machine."
>
> (De Montmollin, 1986)

> "The definition of user interface can be kept simple: it is those parts of the system or machine that the user may come into contact with, physically or perceptually."
>
> (Payne, 1985)

> "We shall define an interface as a physical/conceptual structure both channelling and facilitating communication."
>
> (Martin, 1973b)

> "The interaction between the user and the computer occurs at the computer's "user interface." The designer usually thinks of the user interface as the terminal hardware plus the software that receives, interprets, and sends messages (and displays) to the user. This definition may satisfy the system designer and implementer, but it is inadequate from the user's point of view. Psychologically, the user interface is any part of the system that the user comes in contact with -- either physically, perceptually, or conceptually. (...) The whole conceptual organization of the computer system from the user's point of view -- the user's conceptual model of the system -- is an integral part of the user interface. (...) It follows then that the system's user interface includes not only its behavior at the terminal, but the training and documentation as well."
>
> (Moran, 1981a)

Clearly, the concept of the interface is one which varies substantially from one author to another. De Montmollin provides us with the most traditional, but also the least complete definition, since the interface is defined almost exclusively in terms of physical support. Payne attempts a simple definition, but it remains too general and considers the interface strictly as a part of the computer. Martin correctly points to the communications aspect of the interface, but his definition reduces it to a single dimension. Moran's definition is more satisfactory, albeit somewhat unorthodox for computer scientists, and he has captured the approach of cognitive ergonomics. Obviously the interface is the keyboard, the mouse, the screen and so on, but it is also the content of the screen, the structure of the menus, the error messages, the command syntax and semantics, etc. It is, moreover, also the mental model that the user has of the system, a model which, as we shall see, is rarely complete (Gentner and Gentner, 1983; Gentner and Stevens, 1983; Halasz and Moran, 1983; Young, 1981). The interface also includes the structure of the tasks to be performed, from the point of view of problem solving, in other words, the most appropriate strategies on which the user can draw as a function of his skill and goals. One can therefore talk of an **"objective" interface**, which the designer designs and which is linked to the system itself, and a **"subjective" interface**, which the user engenders and which is therefore primarily dependent on the user's psychological functioning and the context in which his interaction with the computer occurs.

Such a definition obviously expands the concept of the interface considerably, to the point where some authors suggest distinctions similar to the ones we have put forward (Edmonds, 1982; Payne, 1985). Should these psychological considerations be included in what is termed the user interface or not? The question is still open, but it is certain that this dimension, whether it is viewed as an element or a result of the interface, constitutes an essential aspect of the user-friendliness of systems and software. It is therefore the definition that we shall adopt here.

Having said that, it seems to us dangerous to limit the field of cognitive ergonomics to the study of the interface alone, even in its broadest definition, since this is perhaps the wrong way to state the problem. Clearly, the improvement of the interface is a determining factor, but what needs to be optimized is the performance of the entire **user-computer system**, which includes, but is not limited to, the interface. The interface is the visible manifestation of the interaction which embraces a user using a computer system to perform a task or a set of tasks. It is therefore a complex and multivariate interaction, and we need to understand better the players and their relationships with each other.

2.2 The user-computer system

It seemed worthwhile, in order to place the research in cognitive ergonomics in its proper context, to illustrate diagramatically the elements which come into play during the user-computer interaction. Figure 2 shows the three elements of the system: the computer system, the task and the user. Each of these elements can itself be broken down into several dimensions, which we will describe briefly here before examining them in greater detail in succeeding chapters.

The computer system

The computer system can be regarded as including three concentric fields, each of which is based on its predecessors: the basic hardware, the interface hardware and the interface software. The **basic hardware** is at the heart of the system; it is the part of the material with which the user does not come in direct contact, but which nevertheless plays a determining role by limiting the performance of the other fields. These are the physical features of the system (active and passive memory, processor capacity, BUS capacity, etc.) and its processing capacity (number of operations per second, disk access speed, etc).

The user interacts with the **interface hardware** to command the system and receive its responses. This is the input (keyboard, mouse, pen, touchscreen, microphone for speech recognition, etc) and output hardware

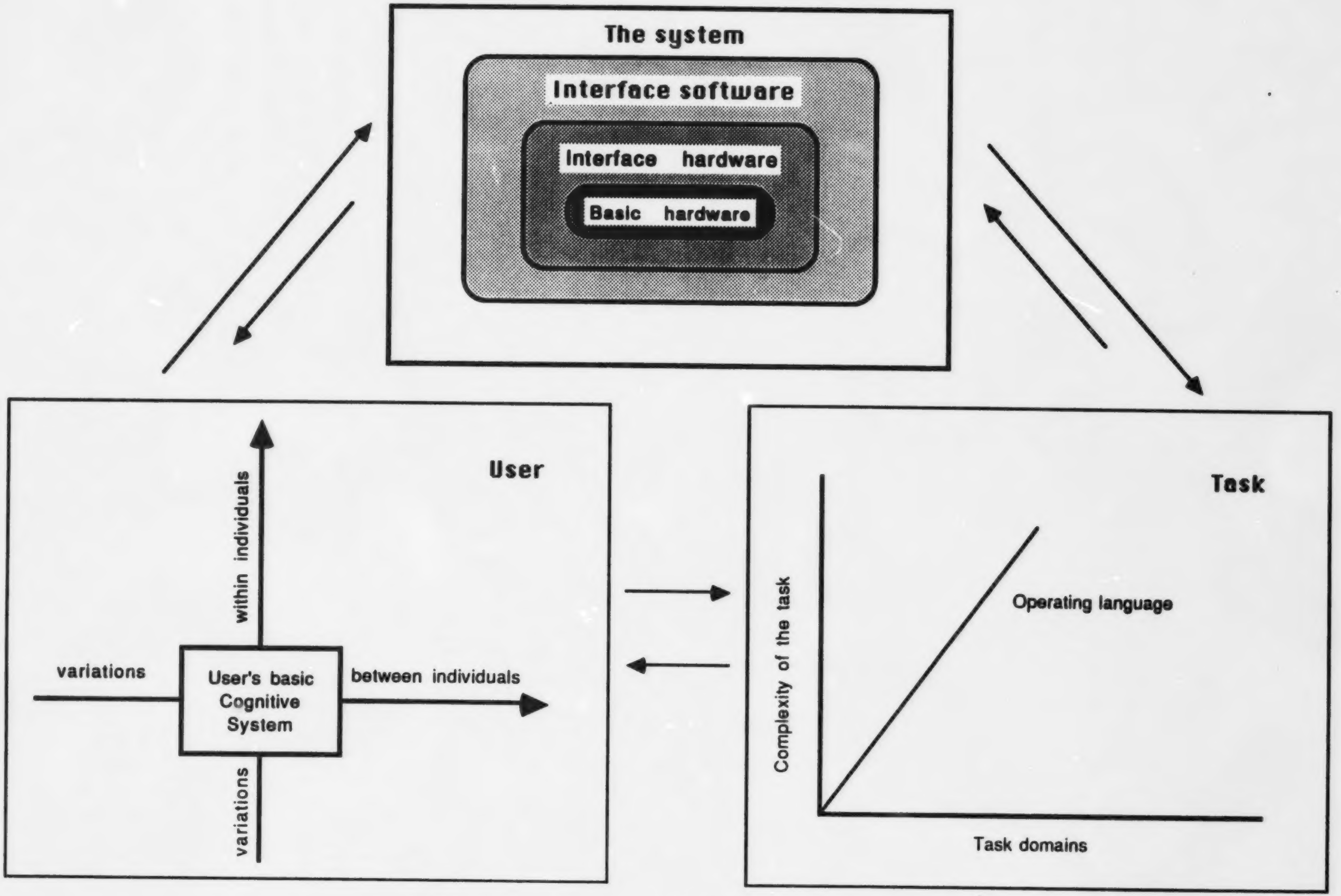


Figure 2 - Model of the user-computer system

(screen resolution, size and color, sound, printer, etc.), but peripherals such as diskettes, disk drives, text scanners, etc. must also be included.

The **interface software** is indubitably the most visible and easily manipulated element of the computer system. Designers generally focus on it and it is the most highly developed aspect of cognitive ergonomics. The interface software is the operating system as it is viewed by the user, it can vary in its complexity and power and whether it can simultaneously support several users or perform several tasks. It is also the way the user interacts with the operators and the subjects of the work: when talking of ways to input commands, we speak of command interfaces, menus, direct manipulation, multimedia interfaces, natural language interfaces, query-by-example interfaces, etc. In terms of display and the manipulation of the work environment, we talk of display by line, screen or object, single or multiple display surface, etc.

The task

The task is often underestimated in evaluating the relative importance of system components. For example, it is often taken for granted that the user is familiar with the task domain and that problems are limited to the interaction with the software. The task is nonetheless the raison d'etre for the interaction, and its characteristics impose very strong constraints on the preferred style of interaction. We draw a distinction between three sources of variations: variations in the task domain, in the complexity of the task and in the mode of operation in the task space.

In terms of the **task domain**, it is clear that the characteristic features of the interaction will vary, depending on whether it involves word or graphics processing, electronic mail or the retrieval of information from a database, or whether the software in oriented towards a specific use (such as word processing) or an integrated environment (such as Lotus 1-2-3).

The **complexity of the task** may vary with the domain, but it also changes within a domain. We are familiar with problems of scale in the area of databases, where the extent of the database becomes an essential factor in determining the best method of retrieval. Similarly, the nature of the request may necessitate a simple command or the performance of a sequence of complex functions. In word processing, the problem will not be of the same order of magnitude if a word is to be inserted or if conditional modifications are to be made regarding the manipulation of the format of a text several dozen pages long. It will be apparent that complex tasks require planning activities which may or may not be assisted by the system interface software and by the operating language of the task software.

The **operating mode or language in the task space** refers to the operating semantics and syntax that the user has to manipulate in order to perform the task. Starting from problem state X in order to reach desired state Y, the user has at his/her disposal a set of commands or operations which enable him to proceed in the task space. In this context, it is important to distinguish between the machine's general operating software and software dedicated to the specific task.

Similarly, a distinction must be drawn between the interaction mode (commands or menus, etc.) of the interaction semantics, which is specific to the task and often to the software used. For example, several databases are available for the Macintosh which operate in accordance with the machine's general environment, but which use quite different operating languages to store and retrieve information. This software is designed for the same machine and the same user interface, but it is possible to demonstrate that their operating languages significantly alter the user's task (Ingenito, in preparation).

The user

The user is obviously a key factor, whose importance, although recognized, is difficult to assess. As we pointed out earlier, today's user has become so variable that it is impossible to describe him. A distinction can be drawn between two main types of variation: interindividual and intraindividual variations.

Interindividual variations are related to the personal characteristics which distinguish users from one another, whether they be physical or psychological: physical handicaps, manual and intellectual skills, motivation, cognitive style, etc. **Intraindividual variations**, on the other hand, arise in the same user as a result of the passage of time or practice: learning, acquisition of routine skills, proactive and retroactive interference, stress and fatigue, etc.

Although these variations are often confused when comparing two users, it is important to distinguish between them. For example, expert and novice users are often contrasted, as if the typology were simple to establish and manipulate. The "expertise" variable is not unidimensional and is definitely not dichotomic, while the question must be asked whether these two extremes are solely the result of practice (intraindividual variation) or if they might also be the result of interindividual variations such as motivation and cognitive style.

Historically, it was long believed that systems could be designed for a typical user which did not vary in either of these two dimensions. Subsequently, attention was focused on one or the other, but without seeing them both simultaneously. Current thinking goes in the direction of an interaction which takes into account both the user's individual characteristics and his/her current situation in relation to the computer system and the task domain.

The interaction of task, user and system

The schematic representation of interaction given in Figure 2 shows the three elements outlined above. But the user, the task and the computer system are in a process of interaction, which is not well illustrated by the directional arrows in the diagram. It is clear that this interaction adds enormously to the complexity, since it moves in two directions:

1) "Real time" interaction, meaning exchanges between the partners. At this level, interaction is conceived of as a dialogue in which the user and the system are interlocutors who interact in a way which is hopefully cooperative about a content (the task) by establishing a common language;

2) Interaction in a more methodological sense, in other words, the mutual interplay of dimensions with each other. The question might be asked, for example, whether a "command" interface is preferable to a "menu" interface, but the answer does not lie in the computer system alone. The selection at the level of the first element must be modulated as a function of the user's skill (second element) and of the task (word processing or electronic mail, third element).

One could cite many more examples of interrelationships between system, task and user. The system's power and software determine the tasks which can be computerized. In return, systems are now often designed to meet the requirements of the tasks they are intended to perform. The development of specialized machines for the LISP language for research in artificial intelligence is one example of this.

As the pool of users grows, new tasks are computerized, tasks which themselves contribute to increasing the variety of potential users. As far as the connections between systems and users are concerned, it is clear that systems are increasingly designed as a function of a specific target clientele and that the development of systems is dependent on their acceptance by users.

In the following chapters we will re-examine in turn each element of the model to compile a picture of the current status of research. First, however, we will examine the status of the question that goes to the very heart of the science of ergonomics: the optimal interaction between user and computer.

2.3 The criteria of optimal interaction

With the proliferation of microcomputers in business and home use, the concept of user-friendliness has become a key aspect of advertising. Common usage defines a user-friendly system as one "which helps a person to perform a task in a natural way, which is easy to understand and use" (Stevens, 1983). This is already a less restrictive interpretation than the one that confuses friendliness and affability. At the University of Montreal, for example, the editor provided for faculty and students is incredibly obsolete, but the operating system, in response to the user's identification code, answers "Hello!", calls him by his first name and says "Goodbye" when he logs off. This is a rather crude vision of what is meant by friendliness. While it is true that the establishment of a more personal relationship with the user may be useful, especially for software designed for the general public, such as videotex databases or some medical assistance software, in which some of the questions asked might seem indiscrete (Lucas, Card, Knill-Jones, Watkinson and Crean, 1976, in Stevens, 1983), friendliness cannot be limited to these minor aspects of the code of polite human conversation.

Burch (1984) proposes a concept of "transparency" as the ultimate objective of user-friendliness. A system is transparent when its design is so well suited to the way in which the user thinks, speaks, memorizes, perceives, processes information and solves problems that the system itself becomes "invisible" because it requires no attention on the user's part. The whole problem, clearly, lies in turning this concept of an ideal system into reality and determining precisely which characteristics of the system would enable this optimal interaction to occur.

In fact it may well not be possible (or even desirable) to define the friendliness of a system in terms of its intrinsic features, whether they involve hardware or software. What is important are the consequences of these characteristics for the efficiency of the individual-computer system, which may be measured by various performance criteria. In the final analysis, these are the parameters which should be optimalized, since they are the ones which reflect friendliness most concretely. We are, however, touching on one of the thorniest problems of cognitive ergonomics -- the search for a clear, functional definition of the objective, in other words, a definition of optimal interaction. The current status of research does not, unfortunately, provide an answer, as is reflected, moreover, in the important report of the committee commissioned by the United States National Research Council to evaluate research needs in the area of human factors. One of the committee's main conclusions reads as follows:

> "The primary research recommendations in the areas of users, tasks, hardware, software, and documentation include a major emphasis on developing new methodologies to evaluate what is meant by ease of use in human-computer interaction. Does ease of use mean the extent to which it is easy to learn to use a computer; does it imply good design of hardware and software for a variety of naive, casual and professional users; does it mean that any task can be done quickly and without errors; does it encompass a component of judged satisfaction about use; or does it mean all of these?"
>
> "Research Needs for Human Factors," Report of the Committee on Human Factors, National Research Council of the United States, Washington, 1983, p. 114.

Every researcher, of course, has his own concept of what is meant by ease of use, which is reflected directly in his choice of the dependent variables studied. Generally speaking, however, these are "local" definitions, which deal with a limited aspect of performance without attempting to obtain a broader measurement of the overall performance of the human-computer system. In the present state of our knowledge, we can do no more than enumerate the criteria which are used most frequently to measure the quality of this interaction. We will list ten which, as will become apparent, do not all differ greatly from those of more traditional ergonomics. They are, however, better suited to multifunctionality and to the more interactive and more cognitive nature of the user interface. Some of them are taken from Moran (1981a), although we have added several others.

Performance time

This is clearly one of the prime criteria of productivity. A user-friendly system should enable the user to perform his tasks rapidly. Caution is indicated here, however: we are talking about the speed with which the task is performed, not the speed with which the system responds. The optimal system is not necessarily the one which responds to the user instantaneously. For example, in large shared-time systems, the designers often at the goal of constantly minimizing the system's response time; when the system was not in heavy use, the wait was almost zero, but it varied greatly at peak periods. It quickly became apparent that under such circumstances, it was preferable to achieve a slower, albeit constant response time, so that the user could benefit from the anticipated waiting period to perform other mental or physical operations. This conclusion cannot, however, be generalized to all situations and all users, and this problem with the effect of time and the variability of response time on performance would appear by itself to be extremely complex (see the discussion in Nickerson, 1986).

Similarly, technical specifications such as MOPS (millions of operations per second) have a definite impact on the power and speed with which commands are executed, but can have only a minor effect on the speed with which text can be edited or information retrieved from a database. In areas such as office automation, in which the computer is primarily a working tool (or a partner), speed of execution depends on the structure of execution of the tasks (the algorithmic or heuristic strategies which are possible and effectively utilized) and the user's mastery of it. For example, we analysed the relational database software sold commercially for the Macintosh, and it is clear that software which appears to have the same interface imposes execution procedures on the user which are quite different and which have an impact on the speed with which the desired information can be retrieved (Ingenito and Giroux, in preparation). Roberts and Moran (Roberts, 1979; Roberts and Moran, 1982, 1983) have evaluated the performance times of various types of text editing software.

Error rate

The significance of errors in the human-computer system can never be stressed enough. Except when their consequences are annoying, such as the irremediable deletion of files or the printing of erroneous calculations, there is a tendency to regard errors as an integral part of any normal interaction. While there is nothing wrong with this, the old adage that "to err is human" should not be applied too quickly to the use of computers. Of course, some errors may be attributable to the user's perception or motor functions (keying error, misreading of the screen, etc.). Even in such cases, however, the error is not always independent of the system. On the contrary, as we pointed out earlier, an operator who misreads an error message or a menu may have made that mistake because the information presented to him was unexpected in terms of his model of the task. Considering "perception errors" as such, is to assume implicitly that the process of perception is guided entirely by data, whereas we know from cognitive psychology that it is also (and primarily) guided by concepts, in other words, in accordance with the subject's expectations (Lindsay and Norman, 1977; Minsky, 1975; Rumelhart, 1975; Schank, 1972; Schank and Abelson, 1977).

Similarly, keystroke errors may be attributable to inattention on the part of the user or to the configuration of the keyboard (the ergonomic inadequacy of the QWERTY keyboard and its French equivalent, the AZERTY keyboard, is well known), but they often result from a poor key-action relationship (particularly in the case of function keys; see John, Rosenbloom and Newell, 1985) or the lack of keyboard standardization (anyone who has used a number of keyboards in recent years will understand this comment).

The "surface" problems, however, which we have just listed account for only a relatively small proportion of the productivity variance due to errors. The sources of errors are many and complex, and this is doubtless why astonishingly little work has been done on errors made in using software. Even Card et al. (1983), in their in-depth study of text editing, base the majority of their analyses on tests performed without error. Nevertheless, and by their own admission, errors account for 26 percent of working time, even though in their case they were dealing with subjects who were experts and were very familiar with the editor being tested. They point out, furthermore, that not all errors are similar in nature, and that a distinction must be drawn between routine errors and those which require problem-solving activity:

> "Although these big errors were method failures, they were classified separately because their times (43,52 and 121 sec) were an order of magnitude larger than simple (4 sec) method-failure errors. The important characteristic of these errors is that their correction involved real problem-solving activity, mostly having to do with the user finding her place in a large text file. These results suggest two radically different sorts of errors that system designers should consider: The first are small, routine errors that can be corrected quickly in a skilled manner. The second are big, infrequent but enormously time-consuming errors that require problem solving to correct".
>
> (Card et al., 1983, p. 179)

It should be remembered that the task in question was the simple one of correcting a manuscript, performed by a skilled operator. It is therefore probable that errors account for an even larger percentage of the time when the user is less skilled or the task is more complex. The distinction drawn by Card et al. is no less relevant for all that: errors take on a disproportionate importance when the user does not know a simple way of correcting them. This conclusion leads us to our third criterion: error correction.

Error correction

While it is necessary to attempt to reduce the number of errors, it would nevertheless be illusory to hope that errors will never be made. The system should obviously guard against the most serious damage by attempting to avoid them (write-protecting software and applications, protection against system failure, etc.) and provide for ways of "recovery" for the most common errors ("undo" type commands). These are however not the most serious errors, the ones which stump and discourage the user. For example, the use of databases with tree structures to run the Telidon videotex protocol proved a partial failure, among other things because in the case of error, users who were unfamilar with the database were unable to "find their place" in the tree, often leaving them fumbling their way around the system. Similarly, inexperienced users of word processing systems have a tendency to rely on the analogy with a typewriter and thus make sometimes costly errors (incorrect paragraph indentation, deletion of a file, etc.).

Those errors which require problem solving on the part of the user are the ones which must be addressed and current work on help systems is intended to do so (Caroll and Kay, 1985; Chin, 1986; Fisher et al. 1985; McKendree and Caroll, 1986; Nakatani et al., 1986). The problem is a double one: on the one hand, the problem must be recognized and diagnosed (which the user is generally unable to do) and on the other, the system must offer possible solutions. The sources of errors are diverse and often difficult to anticipate without a theory of the user's mental model and hence some knowledge of the use of the system in a real situation. This is the reasoning behind the attention being given to process analysis of tasks and knowledge-based interfaces.

Speed and ease of learning

Ease of learning offers obvious advantages in the workplace and is also used as an argument in selling "user-friendly" systems. Three comments are, however, germane.

First, the importance of this factor varies enormously depending on the context. A computer professional will tolerate a relatively long learning period, while office staff will want fast learning and the average user accessing a database in a hotel lobby will not want to spend more than a few minutes doing so.

Secondly, as we shall see later, ease of learning is almost always achieved at the cost of some compromise in terms of software power.

Thirdly, great care is necessary with regard to learning: several studies have shown that even users who have had systematic training are able to master only a portion of the functions of the system they use (Eason, 1984; Folley and Williges, 1982; Nickerson, 1981; Rosson, 1983). The mental model which users develop of the operation of the machine is often inadequate (Bourdeau, 1986; Halasz and Moran, 1983; Young, 1981) and, in order to save memory, they will prefer to repeat simple commands several times rather than learn a more powerful but more complicated and less frequently used command.

Finally, systems like the Macintosh, which appear possible to learn by exploration alone (Robert, 1986a), conceal a trap which we have observed at first hand on several occasions in our laboratory. While it is true that users learn to use this machine quite quickly for simple tasks, they then no longer take the trouble to consult the manuals and subsequently commit costly errors because of failure to master certain concepts. In this case there is a tendency towards excessive simplification in relation to the actual technical complexity of the machine. Learning a computer system is not generally a question of all or nothing and one must be sure that the user is aware of all the possibilities that can be useful (see the usability criterion in a subsequent section).

Learning retention

Learning software is often an intensive process, especially when it occurs in the workplace and in the context of employee training sessions. Learning is then measured a short time, if not immediately, after the completion of training, thus measuring short-term learning rather than longer-term retention of the material learned. This procedure, which is also very common in experimental comparisons of the qualities of different software, does not cause too many problems in the case of software which the user uses on a daily basis, since in such cases the question of retention does not arise. The same is not, however, true of software for less frequently used applications, where it is essential that the user not have to refamiliarize himself with the commands for each new use. Readers will recall Norman's famous critique of the UNIX operating system (1981; see also the reply by DeLeon, Harris and Evens, 1983).

The interface must therefore be provided with features which encourage retention or which at least enable previously acquired knowledge to be retrieved easily. In the case of the names of commands, for example, a coherent, natural vocabulary is needed; care must be taken to ensure maximum similarity between the name of the command and the corresponding action (Barnard, Hammond, MacLean and Mortin, 1982; Black and Moran, 1982; Rosenberg, 1982, 1983; Scapin, 1982; see also Bewley et al., 1983; and Hemenway, 1982, with regard to icons).

Transferability of learning

The astonishing development of hardware and software quickly condemns systems to rapid obsolescence, and workers must accustom themselves to more or less constant retraining. It would therefore be desirable for interfaces to evolve towards a form of standardization allowing for rapid adaptation to change. The configuration of modern automobiles is a good example of a system which has attained a degree of maturity and in which learning is transferable to a high degree. Unfortunately, computers have not yet reached that stage, which is a good thing in another way because standardization often constitutes a significant brake on technological development (see the case of the QWERTY keyboard in the following chapter). An example of the other approach is the success of the IBM-PC family, which has imposed its standard even though the interfaces created for this range of machines are generally less advanced than recent developments.

In the absence of the ability to provide training in a broad range of software on different machines, many people have focused on the learning methods and forms likely to encourage the transfer of knowledge from one system to another and minimize the problems of proactive interference (knowledge of A harms subsequent learning of B) and retroactive interference (in which learning B diminishes performance on A). Examples of such work are those on the conceptual learning of word processing (Bourdeau, 1986; Douglas and Moran, 1983) and programming (Mayer, 1975, 1976), which aim to teach a general model of these operations, independent of the software or the language used.

Mention must also be made of another type of learning transferability, one found within a computer system: this is the consistency of the elements of the interface. The concept of consistency is still only poorly defined (Robert, 1986b) and is understood better by its opposite: a system is inconsistent "if the user exposed to certain aspects of its behavior is led to form expectations about other aspects, which the machine itself then violates" (Young, 1981).

Consistency may take a variety of forms: consistency of the spatial position of the menus (Teitelbaum and Granda, 1983), of the syntax of commands (Barnard, Hammond, Morton and Long, 1981), of their semantics, etc. Consistency offers obvious advantages: quicker learning, easier retention, fewer errors, etc. It is doubtless to promote consistency that companies like Apple impose constraints on the designers of Macintosh software regarding the use of menus and windows.

Functionality of the system

The concept of the functionality of the human-computer system, introduced by Moran (1981a), is its capability of performing a wide range of tasks. The current trends towards integrated systems and local area networks reflect these concerns. Although this is of course not strictly an ergonomic criterion, any evaluation of the interface which does not take into account the real power of the system to assist the user in his work will remain incomplete. Other external criteria, such as the cost of acquisition and operation and the quality of outputs should also play a role in the evaluation of interfaces when a concrete choice has to be made between several alternatives.

Flexibility of the system

Ramsey and Grimes (1983) talk of flexibility in designing a system with the following qualities: 1) applicable to a wide range of tasks, 2) allowing the user several methods for performing a task, 3) allowing the user several ways of invoking commands, 4) adaptable to various kinds of users, especially to varying levels of expertise, and 5) allowing the users to personalize his interaction with the system in terms of his own preferences. Eason (1980) also mentions flexibility, stressing in particular the need to retain the open structure of tasks rather than making them closed, in other words, tightly circumscribed by the system and imposed on the user, who must then submit entirely to the predefined structure.

Flexibility thus consists above all in allowing the user to control his/her own mode of interaction with the system and the way in which he/she performs the task. While this is obviously a desirable goal, it is not a working criterion which is capable of articulating its own problems: offering several alternatives to the user involves at the same time forcing the user to make choices which he/she is not always capable of making, let alone evaluating. Flexibility can thus become synonymous with complexity for the user, or at least the appearance thereof. This is why Ramsey and Grimes report on a number of studies intended to transfer the initiative for dialoguing from the system to the user as he progresses, so that the novice user is initially offered a highly structured environment and interaction, gradually assuming greater control over the type of interaction desired.

Acceptability of the system

The acceptability of the system refers to the reluctance on the part of potential users to using it and to attitudes developed in use. In the broadest sense, acceptability depends on a host of factors, not all of which are related to cognitive ergonomics. The context in which the system is installed and used, the organizational climate and motivation level of the employees are all variables capable of determining whether the system will be well received and sometimes even whether it will be installed. On this basis, "cognitive" acceptability is by no means a negligible quality and results in large measure from other factors dealt with in this section (speed, error rate and recovery, etc). Nickerson (1981) gives several examples of frequent complaints lodged by users regarding response time, functionality, documentation, command language, consistency, etc.

Acceptability stems from these factors, but does not correspond exactly to the sum thereof. Malone, among others, has focused on the motivational aspect of interfaces (Malone, 1980, 1981a, 1981b, 1982) and highlighted several principles which have received little attention but which should be the subject of more extensive research: providing clearly defined goals, intelligent feedback on performance, etc. He further suggests that the interface be equipped with multiple levels of complexity and specifically a striving for "an optimal level of informational complexity." This final recommendation is perhaps the most fundamental. We know that excessively routine tasks cause a drop in the level of motivation and that tasks which overload the cognitive system result in stress and fatigue. In a system which is in daily use, it is therefore crucial that the interface maintain a satisfying level of motivation for the user.

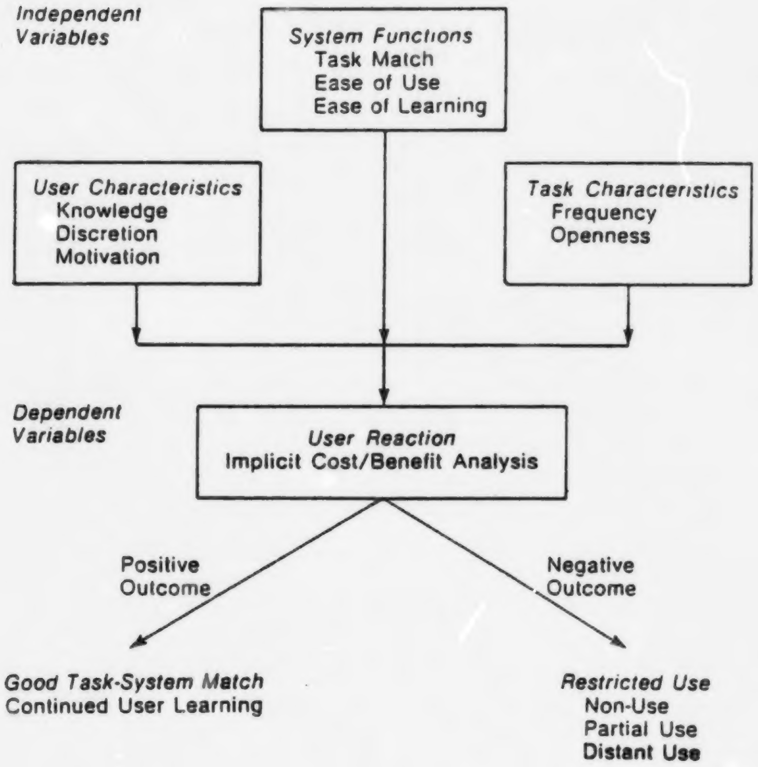
Usability

"Usability," like acceptability and flexibility, is not a unidimensional measurement of user-friendliness, but emerges from the interaction of several factors. As redefined by Eason (1984), "usability" is the **distance separating the potential usefulness of a system from the actual use made of it.** The concept as formulated is quite new, little studied and fascinating. It is easy to see how most computer systems are underused -- and we are not talking here of used in terms of CPU time. Where users have to use the machine in the context of their jobs, underutilization is observed in a high proportion of the available range of applications and commands. When the individual has a degree of freedom in his use of the system, rejection reactions are observed which range through delegated use (responsibility for the task is delegated to another person; see Nickerson, 1981) all the way to total discontinuation of use.

We noted earlier several studies which show that even expert users often fail to take the trouble to learn and/or use certain commands. Eason (1984) for example studies the case of a bank management system in which the tellers have 36 ways to obtain information on a customer's account. Each of the coded commands is intended to examine the customer's transactions from a different perspective (account balance, previous transactions, last withdrawal, last deposit, etc.). Eason handed the employees a series of requests using each code and found that four codes were used for 75 percent of all searches. Subjects frequently succeeded in obtaining the correct information, but by using strategies which appeared non-optimal (use of a series of simple commands, sorting from a more complex list than required, etc.). On the strength of these findings, Eason develops a causal model of "usability" based on the interaction of variables related to the user, the task and the system. Figure 3 illustrates the model thus constructed.

For Eason, the "usability" of systems must be placed in its real context, that is, as a contingency table of the type "a system needs X features for Y type users as a function of Z tasks to be performed." The aim is therefore to predict the performance of the interaction, not according to absolute ergonomic criteria, but according to the functions it fulfils for the potential users. According to this model, the user evaluates his own interaction in terms of the cost/benefit ratio as he perceives it and the determining factor is the outcome of this evaluation. Eason continues by showing how the variables which affect "usability" are often ignored or controlled in ergonomic research and how laboratory experiments therefore are unable to predict accurately the actual use made of the systems tested.

Figure 3 - Eason's (1984) "usability" model



Eason's position is interesting because it reflects a degree of realism which is not all that common among researchers (even in applied research!) However, it raises serious questions for anyone wishing to construct a science of human-computer interaction and points to two related problem-areas: the value of ergonomic criteria and the question of compromises between these criteria.

Weighting and interaction of criteria

In order to demonstrate the objectives of cognitive ergonomics more clearly, we have formulated ten performance criteria which can testify to effective interaction. This list is probably incomplete and certainly not definitive. It is, moreover, inadequate as a guide for system designers wrestling with the problem of predicting how their product will be received by users. These criteria have been formulated to some extent in absolute terms, as if they were all of equal importance and independent of each other. Such is unfortunately not the case and this lies at the root of the problem of the transition between theory and practice.

First, the relative importance of these criteria varies enormously depending on the context in which they are used. Speed of execution and error rate are always important factors because they are omnipresent in the interaction. Nevertheless, user tolerance for a slow or complex system will vary depending on the importance of the task which the system enables them to perform. Ease of learning will count for little in an application which is perceived as being highly worthwhile and used frequently, but will be the determining factor if the user is not highly motivated or if the intended use is occasional. Learning retention will be a relatively negligible factor for a user who uses only one system, while it will be crucial if he is required to learn several. The same holds true for transferability. As far as functionality is concerned, a powerful system will be well received by someone whose tasks are diverse, but will probably be too complex and cumbersome for one who merely inputs data. The criterion of acceptability should also be weighted as a function of the user's level of motivation and personality. Finally, usabilility will be less important if the system plays only a secondary role in the user's daily task or if the system's cost/power ratio is low.

These are no more than a few, quickly sketched examples of the weighting process which should be undertaken by anyone seeking to predict the effectiveness of the interaction as perceived by the user.

Moreover, these criteria are not only not independent, they are often opposed. As Norman points out on several occasions (1983; Norman and Draper, 1986), "There are no simple answers, only trade-offs." Speed and error rate are often inversely related. The features which make a system easy to learn often become drawbacks for the more experienced user. A system of original commands will facilitate retention, but reduce transferability. An integrated, multifunctional system will be highly functional, but also more complex and less consistent. One could go on ad infinitum listing the inevitable compromises, and not only in terms of these general criteria: almost every design choice, down to the smallest detail, results from one or more compromises.

So what now? Well, we have to make do with what we have, which is why practitioners who are aware of the problems inherent in the interface (of whom there are still far too few) are desperately searching for guidelines to follow and theories to apply. Nevertheless, as Norman puts it reassuringly:

> "It might be useful to point out that although there may not be any best solution to a problem in which the need of different parts conflict, there is a worst solution. And even if no design is "best" along all dimensions, some designs are clearly better than others -- along all dimensions. It clearly is possible to design a bad system. Equally, it is possible to avoid bad design."
>
> (Norman, 1983; see Norman and Draper, 1986, p. 56)

Chapter 3.

The system

It is appropriate to begin our analysis of the general model of human-computer interactions by focusing on computer systems, since these are the elements which delineate the possibilities of interaction offered to the user. From the perspective of cognitive ergonomics, the most interesting possibilities for interaction are those defined by the software used. Clearly, the ideal type of interaction and the cognitive skills required will differ depending on whether the software is intended to facilitate graphic design or decision-making. This aspect of software will be covered in Chapter 5, "The Task." There is, however, another aspect of software which, without being totally independent of the task, affects more directly the nature of the interactions: this is the interface software in its more general sense. Thus, whether to correct a text or search for information in a database, interactions can occur using verbal or spatial symbols. In the case of verbal symbols, they may be similar to those used in natural language, but they may also be highly artificial, expressed in a command language with its own syntax. These various ways of interacting with the system raise a number of cognitive problems which will be described in the closing section of this chapter.

If we prefer to postpone dealing with problems related to the interface software until the end, it is because the nature of the interface software used is itself often dependent on the physical or material structure of the machine. Physically, a distinction must be drawn between the machine's peripherals, which are dedicated to interactions with the user, and the internal structure of the machine (see Figure 2, Chapter 2). The interface hardware has a direct impact on the nature of the interactions with the user. For example, the terminals without a screen (of the "teleprinter" type) used a few years ago did not allow any interaction based on the manipulation of icon symbols (generally called "objects"). Such interactions are, however, possible with a system equipped with a graphics screen for representing objects and a mouse or light pen to manipulate them. The second section of this chapter will deal with interface hardware. We will review briefly the cognitive problems and knowledge related to various kinds of peripherals for the entry (input) of information by the user and the display (output) of information by the system.

Before beginning our examination, however, it is worth spending a little time on the internal structure of computer systems, since the nature and the development of interfaces, both in terms of hardware and software, depends ultimately on the computing power of the systems. Thus, graphics screens require far more processing power than was possible in the early days of computers.

3.1 The internal structure of computers

Components

Computer components have changed enormously since the advent of the first machines in the early 1940s. Thus, while the first generation computers used valves, second, third and fourth generation computers have been based respectively on the technology of transistors, integrated circuits and VLSI (very large scale integration) circuits. This technological progress notwithstanding, the basic component of computers can be said to have remained unchanged. Put abstractly and slightly simplified, this component can be described as a relay that can be in one of two different states. It is the efficiency of these components that has been enhanced by the development of electronics technology. First, today's computers are obviously more economical as far as the dissipation of energy by each relay is concerned. With regard to computing power, the most important changes have been the speed with which a relay can change state and the speed of communication between the various relays which make up the machine.

The first computers were limited primarily by the time required to switch from one state to the other: in a valve computer, this time was longer by several orders of magnitude than in modern computers. Today, the speed of communication between relays lies between one one hundredth and one tenth of a nanosecond. Although this speed is itself impressive enough, it would appear that the speed limit has not yet been reached. According to Nickerson (1986), computers of the future may be made up of two new types of component, both of which are currently being developed. The first, called the Josephson junction, consists of two superconductors separated by a thin wafer of insulating material. When no voltage is present, the current passes from one superconductor to another and stops when voltage is applied. The Josephson junction can therefore be thought of as a two-state relay, similar to a transistor. The advantage of the Josephson junction, however, is that it can change state in ten picoseconds, which is 1/100 of the time required by a transistor. The Josephson junction is also smaller than a transistor. Its disadvantage is that it requires a temperature of 4.2° Kelvin (the temperature of liquid helium) in order to work, which creates obvious problems.[1]

The second possible change envisaged by Nickerson lies in what is generally called the **optical transistor**. This consists of a crystal located between two mirrors onto which two beams of coherent light are projected. The projection of the second beam enables parts of the first beam to pass through, ranging from all of it to virtually nothing. The optical transistor may thus assume more than two positions, in addition to being a thousand times faster than a conventional transistor. It can also be used with several parallel beams.

As the speed at which relays change state has increased, the transmission speed between them has become the critical factor. The progress made in this area is due in large measure to the miniaturization of the components, which allows them to be placed closer to each other, thereby reducing the transmission distance. This progress could not have been made without the invention of the transistor in 1948, for which Bardeen, Brattain and Shockley were subsequently awarded the Nobel prize. Integrated circuits were introduced in 1959 by Texas Instruments and Fairchild Semiconductor. Over the next decade, the number of transistors on a chip doubled every year. At that rate, Nickerson noted, it would have taken 300 years for the number of components on a chip to equal the number of atoms in the universe, approximately 10^{79}. This rate of growth could obviously not be maintained indefinitely and since 1973 the number of transistors on a chip has increased by only a factor of four every three years, a rate which continues to diminish. Engineers and physicists thus expect that the number of transistors on a chip will reach an insurmountable physical limit, although the exact nature of this barrier is still unknown. This view can nevertheless be regarded as pessimistic when one remembers that as long ago as 1977 a chip a quarter of an inch square contained a microprocessor that was more complex than that used in the early computers. Moreover, the speed at which the signals are transmitted is now 15 centimetres per nanosecond. According to contemporary physics, we are not far from the absolute limit of 30 centimetres per nanosecond which corresponds to the speed of light. Future progress will thus probably not be made on that front.

---

[1] Authors' note: Considerable progress has been made on superconductivity at high temperature since the original version of this report was written. So the disadvantage mentioned may not be true for very long.

Architecture

Some experts believe that future progress will be not so much in the area of the machine's physical components as in their organization. Simplifying again, it can be said that virtually all computers designed to date have the same architecture. They consist of a memory to store the information and a processing unit which performs operations on the information in a sequence. This architecture bears the name of Von Neumann, its designer.

The processor. There can be no doubt that processing speed would increase if several operations could be performed in parallel instead of sequentially. According to some experts, this will be the distinguishing feature of fifth generation computers. Machines already exist which have several processors working in parallel. However, we do not yet know how to make effective use of them. While "traditional" computing theory was well suited to Von Neumann machines, it has very limited applicability when confronted with the problems which arise when several processing units are working simultaneously.

Theoretical research into the properties of parallel architectures is in progress at present. Some of this research is based on neuropsychology (Hinton and Anderson, 1981; McClelland, Rumelhart, and the PDP Research Group, 1986; Rumelhart, McClelland and the PDP Research Group, 1986) and it is easy to understand why. Each of the basic components of the human brain, that is, each neuron, operates at a maximum speed of between one and ten milliseconds, which is very slow in comparison with a transistor. The speed which characterizes human cognition can therefore be derived only from massive parallelism in the operations performed by individual neurons.

Memory. The brain's computing power is also a function of the number of components involved. The human brain contains some 100 billion (10^{11}) neurons, each of which is in contact with more than 100 others on average, which gives approximately 10^{13} synaptic connections. By way of comparison, in 1965, Licklider estimated the quantity of information stored in all the libraries in the world at 10^{15} bits. The capacity of the largest random access memory (RAM) at the time was 10^{7} bits, but some progress has been made since then. However, the maximum density of active memory units is subject to the limits already mentioned. The optimum support at present for passive memory is the optical disk. The maximum density of data on this support, which is defined by the wavelength of light used, has almost been reached. It is, however, easy to use systems consisting of several disks. One such system, manufactured by Teknekron Controls, has a capacity of 2.2×10^{13} bits (see Nickerson, 1986).

The reason for our having stressed the storage capacities of the human brain and of computers in the preceding paragraph is that research in cognitive psychology and artificial intelligence shows that intelligence is not dependent solely on processing capacity but also on an enormous body of knowledge. Sufficient capacity is therefore required to store all this information, which is prodigal in its use of memory space, especially when presented in the form of complex analog symbols such as human speech.

We have seen in this section dealing with the internal organization of computers that there are limits which we can expect to reach in physical terms. However, this section was also intended to show that the computing power already available has quite literally reached astronomical proportions (the Milky Way contains approximately 10^{13} stars). Clearly, therefore, the limits of our present systems are not physical but rather are due to our ignorance in the area of the organization and processing of information.

3.2 The physical interface

In light of the developments of the past few decades, it is surely not mistaken to believe that one day machines will communicate with each other more than with us. It is, moreover, interesting to note that the term "interface" was originally used to describe communication between the various hardware and software components of machines. However, computers would be of far less use to us today if they were unable to communicate with us and the term "interface" refers primarily to problems of communication between humans and computers.

In this section we will briefly review the main types of devices available for exchanging information between humans and computers. We will adopt the classical distinction between input devices, operated by humans and used to communicate information to the computer, and output devices, which return the information to humans.

Data input devices

The keyboard. The keyboard is incontravertibly the oldest, most widely used and best known of all the devices used for inputting data. The fact that the "standard" keyboard is so widespread generally makes us forget that there are many other kinds. These can be grouped into two main categories: chord keyboards which produce a symbol when several keys are pressed simultaneously; this distinguishes them from single-stroke keyboards, on which each key corresponds to a symbol. We know that the time required to press several keys simultaneously increases with the number of keys required. A chord keyboard can therefore be more efficient than a single-stroke keyboard only if the symbols transmitted contain more information than the symbols transmitted by single-stroke keyboards.

This is true, for example, of the machines used by court recorders, on which each symbol may correspond to a word or even an entire phrase, which would require several successive keystrokes on a single-stroke keyboard on which the symbols transmitted are alphanumerical characters and traditional symbols. Since stenography is a special form of writing, the symbols transmitted by the machine are intended ultimately to be transcoded into the alphabet. However, the intelligence required to interpret the symbols transmitted and transcode them is by no means negligible. The automation of the transcoding process is made all the more difficult because the meaning of several stenographic symbols can vary from one individual to another.

This is not the case with the Velotype, a multiple-stroke keyboard currently attracting a great deal of attention in the Netherlands. The symbols transmitted by this keyboard correspond to the syllables in the Dutch language. Since the rules of Dutch orthography are based on a regular correlation between the sounds and sequences of letters, transcoding can be done automatically by a microprocessor built into the Velotype. It is much easier to learn to operate this machine than machines with a standard keyboard. Unfortunately, machines that operate using syllables are hard to adapt to languages in which the rules of orthography do not always reflect the phonological structure of the language, as is the case with English and French.

Single-stroke keyboards vary amongst themselves by virtue of the symbols represented and by the way in which the symbols are distributed. Some keyboards designed for specific purposes, such as numerical keyboards, have fewer keys (symbols) than the alphanumerical keyboards with which typewriters are equipped. Others, used in communications with computers, have function keys as well as keys corresponding to alphanumerical characters.

Generally speaking, the layout of the alphanumerical characters and symbols on the keyboards used in computers is similar to that used on typewriters. This arrangement, which has become standard over the years, is the one which the first commercially produced typewriter had when it was manufactured by Remington at the turn of the century. One story is that Sholes, the inventor of this first typewriter, designed the layout of the keys in order to minimize the likelihood that the typebars would become jammed. If this story is true, the layout, which is still in use today, lost its justification with the advent of today's electric and electronic machines. As early as the 1920s, Dvorak proposed a much more ergonomically sound layout based on considerations of economy and speed of movement. The typing knowledge and models (Kinkead, 1975; Rumelhard and Norman, 1982) developed subsequently have shown that the Dvorak keyboard is superior to the Sholes keyboard, commonly known as QWERTY and its variants, such as the French AZERTY keyboard. The QWERTY keyboard nevertheless not only remains the standard, it has perhaps even become the prototype of (immutable) standards. The history of the keyboard, wonderfully told by Yamada (1980) and repeated by Cooper (1982), would be of merely peripheral interest if it did not provide an exemplary illustration of economic resistance to change, even in the face of ergonomic evidence that clearly favors the change.

Ergonomic research can have a more significant impact as far as the layout of function keys on the keyboard is concerned, since habits in this regard are still not set in stone. For example, Good (1985) studied the functions used most often in word processing before deciding which ones should appear on the keyboard of a new terminal. The layout of the keys was determined so that their proximity would reflect the sequential dependence of the corresponding functions.

The question may be asked as to whether it is worthwhile today spending time on the ergonomic problems related to the keyboard. A great deal of research has already been done on the physical aspects of the keyboard, such as spacing and the pressure required to operate the keys, and so we have valid recommendations on the subject (cf. Alden, Daniels and Kanarck, 1972; Seibel, 1972). Some experts, moreover, believe that the days of keyboards' being used to input data are numbered and that they will be replaced by speech. Human speech would appear to be the only means of transmitting verbal information in all its complexity more efficiently than the keyboard. The problem posed by speech as a method of inputting data is not a physical one involving the recording of sound and the human voice: it is phonological and linguistic. As we shall see in the section on the cognitive interface, word recognition and the understanding of the message still pose major problems for the machine: keyboards are not about to disappear in the immediate future.

Moving and positioning devices. While the keyboard can be used quite efficiently to transmit verbal information, it has major limitations with regard to the entry of spatial information. Special keys are often provided on workstations equipped with a monitor so that the cursor can be moved and positioned on the screen. However, the use of these keys is generally less efficient than the use of specially designed tools.

One such tool is commonly called a "lightpen." Such expressions are most misleading, since the stylus of these devices does not emit any light, but rather captures the light emitted by the beam as it scans the screen. The devices, moreover, do not directly measure the position of the stylus, but deduce it from the time at which the beam passes under the stylus. The resolution power of such devices is therefore limited by the resolution power of the screen. Furthermore, excessively rapid manipulation of the stylus may cause the cursor to jump on the screen, which is counterproductive when the aim is to draw a continuous line in a graphics application.

The problems described above are far less pronounced when an electrostatic stylus is used in conjunction with a digital notepad which records the positions. The notepad is often placed beside the workstation so that its operation does not hide the screen. Contrary to what one might think, it would appear that subjects adapt easily to the act of manipulating the stylus on one surface and seeing the results on another. This technology is well suited to graphics applications and is probably the best method for recording handwriting. As in the case of speech, the problems posed by handwriting as a means of communication with the computer are in the area of the recognition of shapes rather than physical ones. Such problems are particularly complex in the case of continuous handwriting. As far as capital letters are concerned, only a little experience in typing is required before keyboard operation becomes faster (Devoe, 1967).

A different method for moving the cursor on the screen consists of rolling a ball whose movements are recorded and transmitted to the cursor. The ball may be located in a stationary frame (as is the case with a device named "trackball") or it may be integrated into a mobile support (such as the mouse). Such devices are designed primarily for graphics applications. Since they do not ressemble a stylus, they probably require a longer familiarization period. However, research has shown that they soon become quicker to use than some styli (Card, Moran and Newell, 1983).

As Nickerson mentions, the commercial success of the mouse is such that it is hardly surprising that a "cat" has appeared on the market. This device translates the movements of a finger on a touch-sensitive surface into movements of the cursor on the screen. Touchscreens have, in fact, been in existence for some time: obviously, they do not provide the same accuracy as the devices described above, but almost everyone can move their fingers.

For the handicapped who cannot move their limbs, it is possible to record the movements of the eyes. It is also possible to record physiological measurements, although they are often difficult to interpret. A number of existing technologies can also be adapted to the needs of the handicapped. Even though work in this field is not as far advanced as one might wish, the problems of the handicapped in terms of their interactions with the computer have not been ignored (e.g. Buxton, 1986).

Data output devices

Printers. Until recently, the main method by which information was disseminated and preserved was printing. Even though the paper printer is one of the oldest computer peripherals, it is still in widespread use. Obviously, today's printers come in a greater variety of types and can perform a wider range of functions than earlier ones. The first printers used strikers and were thus restricted in the number of characters they could produce. By contrast, it is possible to compose and produce an almost limitless range of characters with the dot-matrix printers that are compatible with many brands of modern personal computer. The quality of dot-matrix printing, however, is inferior to that of good daisy wheel and laser printers. Graphics applications are also possible on dot-matrix printers, but here again the result is of lesser quality than that obtained with a laser printer or a curve tracer.

The advantages of hardcopy over other forms of information display are obvious. The first advantage is that good hardcopy is easier to read than a copy displayed on a cathode screen. The subjects studied by Wright and Lickorish (1983) worked faster and detected a greater proportion of errors inserted in a text when the text was presented to them on paper than on a screen. Other advantages of hardcopy involve portability and accessibility of the information. A document can be read on a bus or alternatively all the pages can be spread out on the living room floor. Screens, on the other hand, are usually extremely limited as far as the quantity of information displayed simultaneously is concerned and they are still somewhat cumbersome to carry around. The great disadvantage of printing in human-computer interactions is its essentially static aspect.

Screens. The majority of microcomputers sold today are equipped with a cathode ray screen similar in several respects to a television screen. Information is displayed by means of an electron beam which sweeps the screen and illuminates its pixels. The length of time the pixels are lit up depends on their chemical composition (phosphorus or other), but is generally shorter than the time required to scan the entire screen. At a sweep rate as low as that of a television set, the successive illumination and extinction of the pixels would cause the picture to flicker unacceptably. To avoid this, the sweep is generally done in two stages. In the first stage, only the uneven rows are swept, followed a moment later by the even rows. The great superiority of screens over printing as far as human-computer interaction is concerned lies in the fact that the information displayed on the screen may theoretically be changed at each sweep.

The resolution of the cathode ray screen depends essentially on the number of pixels. Whereas screens connected to computers were previously used primarily to display verbal information, the advent of bit-map technology has opened the door to graphics applications. Unfortunately, high resolution, color graphics screens are still relatively expensive.

In addition to cathode ray screens, there are several other types of screens that can be connected to a computer. Everyone is familiar with the liquid crystal and luminous diode screens with which some wrist watches are equipped. There are also plasma screens and projection screens and even touchscreens for the visually handicapped. It would take a very long time to list all the research done on the physical parameters of screens and for this the reader is referred to Galitz (1981) and Shurtleff (1980).

It is perhaps nevertheless worthwhile to mention a problem often raised by people who use cathode screens extensively in their daily work. Such people often report excessive stress or fatigue at the end of the day. A study by the U.S. National Institute of Health found that these problems could not be due to the levels of radiation emitted by the screen. Most of the studies linking screens to such symptoms have been criticized on methodological grounds (Sauter, Gottlieb, Jones, Dodson and Rohrer, 1983). It is not that the symptoms do not exist, but rather than they may just as well be due to the environment of the screen or even to an uncomfortable working position. One exception to this is eye fatigue, which can actually be linked to prolonged use of a cathode screen.

Sound. Thus far, our survey of the display devices has been oriented primarily towards the visual aspects. A computer can, however, also be commanded by sound. Since sound messages are essentially ephemeral, use of the auditory mode is not always appropriate. According to Deatherage

(1972), the visual mode is more suited to long, complex messages which have to be retained for subsequent use or which relate to space. The visual mode is also preferable when the user's sound environment is noisy or when his auditory system is already overloaded. The auditory mode is preferable in the reverse circumstances, when the visual mode is overloaded and when the messages require the user's immediate action and attention.

The word "sound" often calls speech to mind. Computers can to some extent synthesize speech. As in the case of automatic perception of speech, the limitations of the automatic production of speech are found at the level of our existing knowledge rather than that of the physical technology involved. We shall take up the problem of speech later in the section on interface software.

The environments

Ramsey and Atwood (1979) did a survey of research into the sensory and motor factors linked to the physical parameters of each of the various kinds of interfaces that we described earlier. Studies on the simultaneous use of several machines are still very rare, despite the fact that we regularly use at least one input device (usually a keyboard) and one output device (usually a cathode screen). A number of workstations in current use are equipped with a second input device, a mouse, and a second output device, a printer. As the preceding sections show, however, these do not by any means exhaust the possibilities. Research is in progress aimed at exploring the possibility of transforming the entire environment into a workstation (or vice versa). By way of an example, we shall describe briefly the Media Room built by the MIT Architecture Machine Group (see Bolt, 1984).

This is a room, one wall of which consists entirely of a rear-projection screen. The user sits comfortably in a swivel armchair placed in the centre of the room. On each arm of the chair there is a joystick and a touch-sensitive mini-surface. Within reach on either side of the chair is a cathode ray screen covered with a transparent, touch-sensitive surface. A microphone is there to record the user's voice and the room is equipped with eight loudspeakers to control the sound environment.

One of the screens displays an image of each of the objects available in the computer-generated representation of the work environment (the computers themselves are hidden in another room). This environment includes among other things a telephone, a calculator, a calendar, books, maps and photographs. Objects can be selected either by using the joystick or by touching the screen. A more detailed picture of the object then appears on the other screen and on the rear-projection screen.

If the calculator appears, calculations can be performed by pressing the keys represented on the screen. The pages of a book can be turned simply by a finger movement on the touch-sensitive mini-surface. A movement from the upper right-hand to the lower left-hand corner turns the page in the direction of the text, while the opposite movement turns it back. The graphics on the projection screen give the impression that the pages are actually turning. Sections of pictures or photographs can be enlarged or reduced using the joystick. Another application allows the user to stroll through Aspen, Colorado. Photographs have been taken of the city at nine-foot intervals and stored on an optical disk. The user has only to move the joystick from the neutral to the forward position to walk along a street, move it to the right to walk down the street to the right, if there is one...and the images glide past as if you were there, or almost.

Another device which can be used in the Media Room can track the position of one of the user's limbs in space. By attaching this device to a finger, it becomes possible to point from a distance at the objects displayed on the projection screen. The room is also equipped with a limited speech recognition capability. This capability is, however, sufficient to enable such ambiguous phrases as "Put it there" to be interpreted correctly when the user points to the appropriate spot. Research is in progress to use the user's line of sight as an indicator of his centre of attention (and perhaps even the dilation of the pupils as an indicator of his interest).

Attempts are also under way to go beyond the two-dimensional framework of images. As we know, the visual impression of three-dimensionality stems in part from the disparity between the images received from each eye. Bolt and his colleagues therefore had the idea of displaying images that were slightly displaced on each of the two successive sweeps of the screen, taking care to obstruct the vision of each eye alternately. Perfect synchronization between the alternation of the eyes and the alternation of the images enabled each eye to receive an image that was sufficiently different from that received by the other eye to perceive three-dimensionality. If, rather than looking directly at the screen, the subject watches its reflection in a mirror, he will have the impression of seeing a three-dimensional object floating in virtual space behind the mirror. This space can become real by using a semi-opaque mirror which allows the subject to see simultaneously what is behind the mirror in addition to the reflected image from the screen situated in front of the mirror. Thus, in passing his arm behind the mirror, the subject can see his real hand approaching the illusory object. Recording the hand movements and the coordinated movement of the image of the object on the screen then gives the subject the impression that he has moved the object. Clearly, apparatus of this kind are still a great hindrance for the subject who, if he believes in the illusions, runs the risk of hurting his hand on the mirror (quite apart from the blinkers he has to wear). The apparatus needed will be reduced considerably in the future when the three-dimensional images are produced by holography. Speaking of weight, there is nothing preventing us from measuring the subject's muscular contractions and requiring that the force exerted to move the illusory objects depends on the nature of the objects themselves. At that point, the term "direct manipulation interface," which is today used, metaphorically, may be used in a perfectly literal sense.

3.3 The cognitive interface

The complaint heard most frequently from ergonomists working on interactions with computers is that the human factor is considered too late. Rather than being able to contribute to the harmonious development of new technologies, the science of ergonomics frequently has to be content with solving the human problems caused by the new technological tools. At this stage, economic pressures may be such that the ergonomists' suggestions are ignored (as the example of the history of the keyboard shows).

Unfortunately, to judge by the preceding paragraphs, this situation will probably persist for several decades more. Our technological knowledge opens boundless fields of opportunity, while our knowledge of the human being is very limited indeed. In the absence of sufficient knowledge of the human factor, economic factors will doubtless dictate which of the opportunities offered by technology will actually be realized.

The fact that ergonomic science is dependent on technological innovations will also be evident in the subsequent paragraphs. Since the dawn of the computer age, we have been able to communicate with computers using languages written specifically for this purpose. We consequently know more about programming languages and command languages than we do about icons and the organization of windows, since these methods of communication (which will be described later in this section) are more recent.

Language-based interfaces

Words and abbreviations. The evolution of the languages available to communicate with the computer reflects a growing flexibility. In the heyday of "machine language," users had to express their instructions and designate the address of the content of the memory by sequences of binary numbers. The advent of "assembler" programs allowed users to use abbreviations to designate operations and memory registers. However, the match between the abbreviations and their interpretation was determined by the machine. This one-to-one match no longer exists in the case of compilers: the words used designate commands which may be executed in various ways by different machines. Progress has been made in so many other areas that it would take too long to describe them here. The important aspect of these developments is that they have made it possible to use virtually any term to designate virtually anything. Since it is nevertheless necessary to choose terms to designate commands, the question asked of software designers is what are the best ones to use?

Several researchers have focused on this problem. Landauer, Galotti and Hartwell (1983) began by analysing the words users would choose spontaneously to designate the five operations (insert, delete, replace, move and transpose) and basic objects (space, character, word, line and paragraph) involved in the task of text editing. Landauer and his colleagues found wide variations in the terms chosen by users to designate each of 25 possible operation-object pairs. According to the data, the probability that two different users would use the same verb in the same situation is estimated at only 8 percent. This finding should not have come as any great surprise to the research community, since one of the properties of natural language is to offer several alternative words to designate the same reality. Another property of natural language is that the same word can often designate different realities. This problem of lexical ambiguity reappears in the results of Landauer et al., where the probability that the same subject will use the same verb in two similar situations is approximately 34 percent.

When confronted with this degree of variability, it is clear that the software designer's intuitions regarding the choice of vocabulary are no longer sufficient. By way of an example, the terms chosen by the software designer used in the experiment by Landauer et al. match those chosen by a majority of users in only one of the 25 possible combinations. Two other approaches have been suggested. The first, which is illustrated by the work of Rosenberg (1982), consists of giving preference to terms which most faithfully describe the operations they designate. In contrast to this system-oriented approach, the second is more user-centred and consists of choosing the terms which are most easily learned and recalled by the software users.

The second approach is illustrated by the work of Black and Moran (1982) at Xerox PARC and of the Barnard group at the MRC Applied Psychology Unit (e.g. Barnard, Hammond, MacLean and Morton, 1982). These authors wondered whether it was better to use general terms or more specific ones. It might be assumed that general terms would be preferable, since they come easily to mind in appropriate situations. On the other hand, it is possible that general terms would also come to mind in inappropriate situations, which would suggest that specific terms are better. The findings of these experiments resulted in improved performance on the part of users who had learned to use the more specific terms. It would therefore seem that the user-centred approach converges with the system-centred one, since terms which are specific to the operations of the system are also the ones best suited to the users.

It is nevertheless worth noting that the results described above were obtained in a situation in which the subjects themselves had to "recall" the command names. The situation is very different in the case of systems equipped with menus, where users merely have to "recognize" the appropriate option from the list displayed. We know that generally speaking a task involving recognition taxes the human memory less than one involving recall. However, the problem with words is not solely one of memory. It is also a problem of the internal consistency of the vocabulary used. Thus, if a word designates an action, users may believe that the opposite action is designated by its antonym, this is not and cannot always be the case. Some systems allow the user to use several different words to designate each command: while this reduces the memory problem by increasing the acceptable vocabulary, it nonetheless does not eliminate the problem of consistency.

Another word-related problem (albeit more prosaic than the previous ones) is that words take a long time to type. Abbreviations may represent a considerable time saving. They are of course less useful when the choice of commands can be made simply by pointing to a menu option. The abbreviations accepted by various systems come in all shapes and lengths. Instead of typing the entire word, it is often possible to stop at the first letter, which is sufficient to distinguish the designated word from the other words in the vocabulary. This method generally produces abbreviations of varying lengths. Another frequently used method of abbreviation is contraction, which consists of eliminating the vowels in words and retaining only some of the consonants. Still another method consists of retaining only the initial syllable of a word. Sometimes abbreviations are not even derived from complete words or, if they are, no one remembers which words they refer to: how many UNIX users can say which word is represented by the abbreviation "cat"?

A great deal of research has been done to determine which of the methods of abbreviation is the best (see Nickerson, 1986). Moses and Ehrenreich (1981) have summarized this work and concluded that contracting words is one of the most effective methods, but that it is preferable to use abbreviations of a set length. Grudin and Barnard (1985) have shown that abbreviations designed on the basis of these principles are learned more easily and result in fewer errors than abbreviations invented spontaneously by users. Another aim of the experiment was to establish the point at which it is useful to teach abbreviations to users. One group of subjects learned the abbreviations in the first session of the experiment, without knowing to which words the abbreviations referred, while a second group learned the words first, with the abbreviations not being introduced until the second session. The second group's performance was better during the second session and subsequently.

Syntax. It is not enough to know the words and abbreviations of a command language in order to use a system efficiently. The user has to know how to arrange correctly the terms which designate the actions and the terms which designate the objects affected by the actions. Therein lies the problem of syntax. In certain natural languages, including French, action verbs precede their object complements, while the opposite is true in others. Some natural languages permit both orders. Artificial languages reflect an equally great -- and perhaps even greater syntactical diversity, since the constraints of order are more arbitrary than is the case with natural languages.

In view of their abstract and arbitrary aspects, command languages present enormous learning difficulties. These difficulties can be reduced if the order of the commands and of their arguments is fixed and systematic. It would also appear that command languages whose structure is closer to that of the user's natural language are easier to learn (Ledgard, Whiteside, Singer and Seymour, 1980; Scapin, 1982). Some experiments (Scapin, 1982) suggest that syntax retention improves when the subjects themselves have chosen the order of the commands and the arguments.

Natural language. One way of avoiding the problem of learning would be to communicate in the users' natural language, since they are already familiar with the vocabulary and syntax. A further advantage of natural language lies in the wealth and flexibility of expression that it allows. The expressive power of a language stems from its capacity to combine. A mere 26 letters are thus sufficient to write all the words in the French language. Obviously, not all combinations of letters produce French words: there are rules of orthography which determine in a more or less rigid way which combinations are acceptable. Similarly, the rules of syntax determine which combinations of words constitute sentences. Nevertheless, the number of sentences that can be expressed by following these rules is literally infinite.

This expressive power also constitutes the major obstacle to the use of natural language in communications with computers. It is possible to load into the computer's memory all the words in a language and even a considerable number of expressions (such as "Hello, what's your name?") which we will refer to as "templates." It is nevertheless theoretically impossible to enter all the possible sentences.

The earliest automatic systems for processing natural language were based on the almost exclusive use of templates to recognize the user's inputs and to generate outputs. One such system is the ELIZA program, designed by Weizenbaum, which played the role of the therapist in a psychological consultation with the user. Weizenbaum (1976) reports that, despite the system's lack of flexibility, the users took the game seriously.

Current systems are no longer based on templates, but use rules which enable them to analyse the structure of a considerable number of sentences. Unfortunately, we still do not know all the rules that would make it possible to analyse all the possible sentences in a language.

Another problem related to natural language is that it is not sufficient to deduce the structure of the sentences. We also have to determine their meaning. However, as we mentioned above, some words are ambiguous: to understand their meaning, not only must the surrounding words also often be considered, but also the general topic of the conversation, the physical context (e.g.: "Put it there") and sometimes even the intentions of the person uttering them. In other words, in order to understand natural language in its entirety, we have to possess and coordinate an enormous body of knowledge, and we still do not know how humans do that.

Even though there is still no general system for recognizing and generating language automatically, there are nevertheless specialized systems which have had a degree of success. Most human-computer interactions today occur in the specific context of well delineated tasks. The computer can therefore possess all the knowledge of the field required to interpret the user's questions and commands and to produce suitable responses. Furthermore, Chapanis, Parrish, Ochsman and Weeks (1977) have shown that users tend to use not only a limited vocabulary but also stereotyped grammatical forms when interacting with computers. Automatic language analysis systems can benefit from these tendencies.

One final obstacle to the use of language as a mode of communication with computers stems from its redundancy. Messages expressed in natural language are generally longer than the corresponding messages expressed in an artificial command language. The user must therefore perform more keystrokes, which increases the execution time. Clearly, this problem is less acute when the messages are uttered orally.

Speech. In addition to being subject to the syntactic and semantic problems described above, phonological problems make automatic analysis of speech very difficult. Whereas the letters that make up a word are distinct from one another, sounds overlap within a word. Printed words are separated by a space, but the sounds in a word often overlap with the next word. The impression that words are separated by silences is an illusion, as can be demonstrated by an attempt to detect the boundary between words in a foreign language. At the moment we do not know the general procedure for segmenting the acoustic chain and extracting the segments that correspond to the individual words.

Most commercial automatic speech recognition systems function only for isolated words or phrases. Such systems must be divided into those intended for use by a single individual and those intended for multiple users. In the former, the user must first pronounce each word and phrase in the vocabulary so that the computer can store a template of the corresponding acoustic segment. This is the learning phase. When in use, the computer compares the words pronounced by the user to each of the templates in its memory by means of a fairly complex matching system known as **dynamic programming**. The success of such systems depends in large measure on the options which are authorized in the vocabulary. The greater the extent to which words have different acoustic representations, the less chance there is that the computer will make an error and recognize a word other than the one pronounced. Systems intended for multiple users operate according to similar principles, but their vocabulary is generally more limited: since the same word can be pronounced in different ways by different people, the options allowed in the vocabulary must be as acoustically distinct as possible.

Analogous, albeit less serious problems are found in automatic speech generation. Although it is very simple to reproduce or even to synthesize predetermined messages, it is very difficult to compose a new message on the basis of the primary sounds in the language. This requires an initial blending of successive sounds followed by the imposition of an intonation contour on the entire message. Although remarkable progress has been made in this field over the past decade, the speech produced still fails to achieve a totally natural quality.

Even though we cannot, with our current knowledge of the automatic synthesis and analysis of speech, build interfaces which reflect the full wealth of language, we can already build interfaces to meet less ambitious goals. It is, for example, already possible to give commands orally to a system, as is illustrated by the Media Room described earlier. Some authors point out that progress in the automatic synthesis and analysis of speech will raise issues which may attain philosophical dimensions. First, if the system expresses itself naturally, people will be inclined to think that it can understand everything they say. However, even if the system can understand all the words uttered, this does not necessarily mean that it can understand their meaning. Secondly, what type of voice would the computer, this partner at work and at play, have: a man's, a woman's, the voice of the user's spouse or someone else's? Finally, should we aim for a faithful copy of human speech, or should the interface retain artificial qualities? This question will be raised in a very concrete form in the not-too-distant future.

Iconic interfaces

The advent of bitmaps has literally revolutionized the interface. Before this technology was available, human-computer interactions usually took the form of a limited dialogue between the user, who entered the commands and the system, which responded. Now the user can select and perform or have the system perform actions on objects represented graphically on the screen, hence the metaphorical term "object manipulation."

Menus. The importance of this change in perspective is illustrated by the appearance of menu interfaces. These interfaces allow the user to select commands from the options displayed spatially on the screen. Since the options can be displayed in the form of words, the menu interfaces are not limited only to systems equipped with bitmaps. Indeed, there are now menu interfaces on traditional systems. The fact that such interfaces appeared after the advent of bitmaps shows, however, that it was first necessary to change our conception of the interaction to a spatial one.

Several researchers (e.g. Ramsey and Grimes, 1983; Shneiderman, 1982) soon realized the importance of this change, with some even going so far as to claim that direct manipulation interfaces would prove a universal panacea. It is true that some problems can be reduced or eliminated in direct manipulation interfaces. Thus, as we mentioned earlier, the display of command names in a menu makes it easier to learn and recall them. Users can even avoid learning the syntax of a complex language when the system itself supplies the options that are acceptable at each stage in the command input process. On the other hand, direct manipulation interfaces raise new problems concerning the organization of space, which is relatively limited on existing screens.

One question that has arisen regarding the organization of menus consists in knowing whether it is preferable to display simultaneously the maximum number of possible options, even at the cost of monopolizing all the space on the screen, or to group the options into categories, even if the user then has to make a second choice from the elements of each category. The first approach is known as "breadth-first" organization, while the second is called "depth-first" organization. Several studies on the question have been carried out (see Landauer and Nachbar, 1985) and it would appear that breadth-first organization is preferable in a number of situations. The fact that various options are displayed simultaneously on the screen allows the user to compare them easily. This process of comparison is obviously facilitated when the layout of the options on the screen is such as to enable the user to locate them directly without having to scan the entire screen with his eyes. The number of successive choices is fewer than in depth-first organization, which saves time, and there are fewer errors. We know that users have a great deal of difficulty in relocating themselves when they make an error in a hierarchical choice sequence. Unfortunately, a perfectly horizontal organization is not always possible. The options supplied in the large databases are often so numerous that they would spill over the four sides of the screen.

Windows and lenses. There are two ways to break through the spatial limits imposed by the size of the screen. The "hard" way consists of increasing the physical size of the screen, as in the Media Room. This solution poses psychophysical problems associated with the resolution of the screen and of the eye. It is also very expensive. The "soft" method consists of increasing the space represented. Thus, rather than thinking of the screen as circumscribing space, it can be viewed as a window opening onto a larger space. Several personal microcomputers, including the Macintosh, already offer the possibility of having several windows open at once. Unfortunately, the windows are superposed on each other on the screen, so that the information displayed in one window may hide that contained in another. The metaphor of the notice board therefore seems more appropriate than that of the window. On other systems, the windows must be thought of as doors through which the user can enter a new space which has just replaced the preceding one on the screen. In sum, as in the case of the menus, there is a breadth-first way to organize windows and a depth-first one.

"Lens" techniques are not yet as widespread as the window systems. In the Media Room, it is possible to choose between a complete, but not detailed picture of a space or object, and an image of the space that is more detailed but also more limited. It is also possible to move in a relatively uninterrupted movement from an overview to a close-up. These situations correspond to the possibilities offered by telephoto, wide-angle and zoom lenses in optics. An intermediate possibility between the close-up and the wide-angle corresponds to the fish-eye lens. This displays in a detailed manner the information that is the focus of attention and progressively reduces the grain in the remaining data as a function of its distance from the centre of attention. All these techniques for using space can be used in graphics applications. They can also be adapted for displaying verbal information (see Furnas, 1986).

Icons. One of the great advantages of bitmap technology is that it makes it possible to represent actual objects or abstract concepts in the form of images of symbols, called icons. The question of the evocative power of these images thus becomes acute and is only beginning to be studied in the field of human-computer interaction. More generally, these are all problems related to imagery and to mental representations of space which have shifted to the foreground since the advent of direct manipulation interfaces.

Chapter 4

The user

As we mentioned earlier, the primary focus of ergonomics is to adapt work to man, the tool to the user. A thorough knowledge of man is therefore a prerequisite and the expression "know the user" is one of the fundamental maxims of ergonomics. However, from the earliest days of ergonomics, this maxim has been interpreted differently depending on the tools and machines under consideration.

The existence of tools designed to assist man in his physical labor caused ergonomics to focus very early on factors related to the manoeuvrability of the tools and the health of their users. During the Second World War, with the advent of machines for detecting signals (such as radar, for example) and operations control, the attention of ergonomics was of necessity extended to man's perceptive functions and attention span. Ergonomics and psychology began to converge on common problems.

The advent of the computer enlarged still further the field of knowledge required in relation to the user. We find today in newspapers and periodicals specializing in ergonomics many exhortations to interpret the maxim "know the user" in such a way as to cover the entire range of human cognitive functions. The advent of the computer has also had a considerable impact in psychology, since man's cognitive functions are now perceived as symbolic information processing systems. This approach is now the dominant one in psychology. In the following section we will also present a brief description of the user based on research into the psychology of information processing.

If the advent of the computer compels us to expand our knowledge of the user, it also enables our knowledge to have a greater impact than in the day of physical ergonomics, when it was unthinkable to want to adapt the machine to each individual user. In fact, adapting a physical machine to different users implies a change in the industrial production process, and such change can rapidly become too costly. This is not always the case with computers: the computer is not limited to manipulating specific symbols in a specific situation. Each computer is a universal machine, capable of imitating any other machine in manipulating symbols. The only change required to do this is a change in the software. Changing a computer's software, however, does not involve an entire industrial infrastructure. As far as cognitive tasks are concerned, therefore, it is entirely conceivable that machines can be adapted to different users, although this requires a more nuanced knowledge of the users themselves.

In the days of physical ergonomics, the knowledge sought in ergonomics was essentially statistical: the aim was to specify the physical properties of the average user and determine the variations in these properties in the population. Anthropometrics is a good example of this type of research. Th parameters obtained could then be used to build machines adapted to the majority of the population. In the psychological context, the study of individual differences was used primarily in personnel selection. Since it was not possible to adapt machines to individuals, the aim was to select from the population those individuals who were best suited to a specific job. Provision also had to be made for relative lengthy training periods. Personnel selection and training thus consisted of adapting the individual to his work.

The computer has the potential to reverse this situation and bring it more into line with the goals of ergonomics. Psychological taxonomies can be used to develop software that is more nuanced and better adapted to various categories of user. In the second section of this chapter we will present a brief overview of research in this field.

This research still leaves us some considerable distance from the individual user, with his personal characteristics and idiosyncrasies. Is it absurd to envisage personalized software? Given the cost of programming, it is obvious that no company could hope to market completely individualized products: nevertheless, it is already possible to market a product that will adapt to the user. Since computers arrived on the scene, informatics and more specifically artificial intelligence have been added to the list of disciplines considering the user. Some research in progress in these disciplines is concentrating on the development of software capable of constructing a model of the user and acting on the basis of the model. This research will be reviewed in the third section of this chapter.

In sum, this chapter will attempt to summarize the status of our knowledge relative to the user by adopting an increasingly nuanced perspective: beginning with the typical user, we will progress to user types and subsequently to the individual user.

4.1 The typical user

For some thirty years now, psychological research has attempted to characterize human cognitive functions as information processing systems. This research has given rise to much controversy and still does today. Nevertheless, from it there has emerged a vision of man which is generally accepted and which we will describe briefly in the following paragraphs. This view is described in detail in several introductory books to cognitive psychology, such as those of Anderson (1985) and Howard (1983). We will limit ourselves here to a rather more simplified vision drawn from Card, Moran and Newell (1983). These authors have attempted to incorporate the main experimental results obtained over the course of the past three decades into a general model capable of describing the performance of the typical user in a variety of situations involving interaction with machines. This means that the complexity of cognitive phenomena has been deliberately sacrificed for the sake of the precision of the model.

The model proposed by Card, Moran and Newell is called the "Model Human Processor." According to this model, humans have three memory systems. On the outer periphery, there are the sensory information registers. As their name implies, these registers preserve the sensory information which comes from the outside world by visual and auditory means. The quantity of information that can be contained in these registers is estimated at approximately 17 letters for the visual register and five letters for the auditory register. The time during which this information can be retained in the registers is very limited: approximately 200 millisecond (ms) for the visual register and 1,500 ms for the auditory register. These are the initial limitations of information processing in humans.

In order not to lose it, information has to be transferred to another memory: working memory. In the model, the operations by which sensory information is transferred to working memory are attributed to the perceptive processor. The information contained in working memory retains its visual or auditory character, but it has been interpreted. One result of this interpretation is that the information is now grouped into significant units (called chunks). Thus, for example, the word "yes" constitutes only one significant unit, whereas a sequence of letters without any meaning such as "bfx" constitutes three units. The total capacity of working memory is approximately seven significant units, which makes telephone numbers very easy to remember. Just as it is possible to remember a telephone number by repeating it over and over again in one's mind, the information contained in working memory will be retained as long as it is the focus of attention. If, however, attention is lacking, the information will be lost at a speed commensurate with the quantity of information contained in the memory. Approximately seven seconds is thus all it takes to forget three significant groups, whereas a single group can be retained for slightly more than a minute without attention.

The cognitive processor operates on the representations contained in working memory to determine what action to take, while a third processor, the motor processor, is responsible for controlling the movements required to perform the task. Each basic operation performed by the processors requires approximately 100 ms, but the exact speed of these operations may vary as a result of various factors, as we shall see later.

To determine what action to take, the cognitive processor may if necessary access the data contained in the third memory system: long-term memory. This memory contains all the acquired knowledge. The knowledge is associated with other knowledge in accordance with semantic interrelationships. The total capacity of long-term memory is to all intents and purposes unlimited and the knowledge stored in it is never lost. However, the fact that all the acquired knowledge is preserved does not mean that it is always accessible. In the model, knowledge is retrieved by association, and it can happen that the associations retrieved do not contain the information sought.

The model includes a set of principles which govern the operation of the various processors and memory systems. Thus, for example, one principle states that the operating speed of the perceptive processor depends on the intensity of stimulation, while another asserts that the operating speed of the cognitive processor is linked to the cognitive complexity of the task. When the complexity of the task depends on the number of possible options, the decision time of the cognitive processor is determined by the uncertainty of the situation, according to Hick-Hyman's law. The motor processor obeys Fitts' law, which states that the duration of the movements is a function of the distance to be covered and the precision required. As far as the memory systems are concerned, one principle states that the accessibility of information depends on the extent to which this information can be distinguished from other information present. Another principle, the encoding specificity principle, states that the best indicators for retrieving knowledge are those that were present when the learning occurred. A final principle links the speed at which a task is performed to the amount of practice.

All these principles and all the parameters of the model are derived from classical experiments in cognitive psychology. In addition to providing a synthesis of acquired knowledge in this field, the model has the additional advantage of being numerically defined. It is thus possible to calculate the effects of certain decisions regarding the design of a machine on the user's future performance. For example, the model can predict the time required to recognize a stimulus, select the correct response from a set of possible responses and perform the appropriate movements. The model even sometimes makes it possible to decide on the relative mnemonicity of various sets of commands.

On the other hand, the model's limitations are evident: it exhausts neither the basic capacities of man nor of the machine. Since the advent of the computer, human-machine interactions can take extremely complex forms going as far as the use of natural language. Card, Moran and Newell's model, however, makes no provision for a mechanism for understanding natural language. More generally, the model is unable to handle situations requiring reflection and comprehension on the part of the user. Card et al. formulated the principle that, in such circumstances, behavior is determined by a set of factors such as the structure of the task, the goal and the operations available to achieve it, the user's knowledge, the information perceived and the limitations of processing. Of these factors, however, only the last two are covered by the Model Human Processor. In other words, to reflect performance in complex situations, models must not only be specific to the task at hand, they must also make provision for the users' levels of ability. This latter requirement is by definition impossible to satisfy in a canonical model of the typical user.

4.2 Types of user

In this section, we will focus on the factors which differentiate users from one another. Since it is possible to distinguish users according to various dimensions, it is hardly surprising that research in this field is marked by heterogeneity. We have grouped the research by the dimension considered. Some of the dimensions enable users to be classified in a relatively stable manner, in the sense that the users, once classified, are unlikely to change category over time. This is the case, for instance, with users' cognitive skills, such as the various verbal and spatial skills which are perceived as being fixed in adult subjects. It is, however, not true of expertise which is specific to a given task: with experience, a novice user will probably change category and become an expert. This section is organized in such a way as to follow a slope in the stability of the dimensions considered, starting with the most stable and moving towards the most variable.

Cognitive skills

Some of the researchers who have studied users' cognitive skills have concentrated on a single skill, while others have attempted to evaluate the relative importance of a range of skills on the subjects' performance. The study of reading skills by Roemer and Chapanis (1982) is a good example of this first type of research. The authors attempted to discover whether this skill influenced the learning of a word processing system. The subjects participating in the study were classified into three groups according to whether their reading skills were weak, average or strong. Three different instructional manuals were used, representing levels of complexity in their style ranging from the fifth grade to the fifteenth year of education. Generally speaking, the weak readers took longer to complete their assigned manuals and made more errors in the practical exercises as well as on the comprehension test. The manual's level of complexity, on the other hand, had no effect on the dependent variables considered, which makes the findings somewhat suspect.

The work of the team of Egan and Gomez at Bell Laboratories also focused on learning word processing. However, these authors considered a range of skills. The results of an initial study (Egan, Bowers and Gomez, 1982) showed that apart from reading skills, the subjects' age and spatial skills also influenced learning. The subjects in this study had to learn to use a line editor (ED). A second study (Egan, Gomez, Wheeler, Sharma and Gruchacz, 1983), examined learning on a full screen editor (TED). Spatial skills remained a preponderant factor in the second study, although it became apparent that the subjects' age had less of an influence than in the previous study. According to these authors, the various skills affect the components of the task of word processing differently. While spatial skills are required to locate relevant information on the screen, the age factor is linked to the complexity of the commands to be performed. The reduced influence of age in the second experiment would thus be attributable to the fact that the TED editor has a more simple command structure, which reduces the cognitive burden on older people.

The approach taken by Egan and Gomez (1984) in their most recent work is also based on an analysis of task components. The difference between this approach and others based on global performance measures is that it makes it possible to determine the cognitive skills required by each component. The results thus obtained have a degree of generality since certain components are also present in other tasks. Such is not the case with studies based on global performance measures: the results of such studies are necessarily limited to a single task, since each task is unique when considered in its totality. Insofar as the nature of the tasks is determined by the present state of technology, the validity of the results obtained in these studies is likely to be somewhat temporary. In all, an approach based on a "componential" analysis of tasks seems to us more likely to provide significant results regarding the cognitive skills required of users. Word processing is, unfortunately, the only aspect of cognitive ergonomics to which this approach, which was inspired by the work of R. Sternberg in differential cognitive psychology, has been applied and even in this limited context the research has concentrated exclusively on the learning phase.

Cognitive styles

An individual's cognitive style characterizes his general approach to his environment. Cognitive style, like cognitive skills, is supposed to be a stable personality characteristic. Cognitive styles are nevertheless different from cognitive skills in that they are not affected by the nature of the information to be processed. Thus, an individual's skills generally vary depending on whether the information to be processed is verbal or spatial in nature, but the individual's cognitive style remains the same in either case.

Classifications based on the cognitive style of individuals generally take the form of dichotomies. The best-known distinction is doubtless that between "field independent" and "field dependent" individuals. The term "field independence" was initially conceived to designate subjects who demonstrated greater facility in a perceptive task consisting of extracting a shape from its context. The expression nevertheless took on a broader meaning in the educational context, in which it is used to designate subjects who exhibit a tendency to analyse and structure the information presented, while field dependent individuals adopt a more global approach to new information.

Apart from field dependence, many other distinctions in cognitive style have been proposed. Thus, in education, Pask (1976) distinguished between two styles of learning: the "serialist" style, which, as the name implies, consists of proceeding by a series of stages, and the "holistic" style, which consists of searching for global comprehension. In the area of decision-making, Kagan, Rosman, Day, Albert and Phillips (1964) suggest a distinction between "impulsive" and more "reflective" subjects. This distinction reflects the speed of decision-making. Another classification proposed by Harvey and his colleagues (Harvey, Hunt and Schroder, 1961) depends more on the quantity of information considered by the subjects. With regard to intelligence, Guilford (1967) distinguished between two types of thought: **convergent** and **divergent** thought. Some neuropsychologists prefer to talk of cerebral hemispheres -- and the list goes on and on.

It seems difficult to reconcile such a proliferation of classifications with the concept of cognitive style. As we mentioned earlier, cognitive style is supposed to be a general personality characteristic. However, while each of the classifications proposed is based on different psychological properties, none of these properties constitutes a truly general personality characteristic. On the contrary, it is possible that the various classifications refer to the same properties, merely under different names. In this case, so-called field independent individuals would also, by definition, be serialist, reflective, convergent, etc., while field dependent individuals would be holistic, impulsive, divergent and so on.

If it is true that cognitive style represents a general way of interacting with the environment, it will probably also be a preponderant factor in determining the relationship to technological tools. Van der Veer's team in the Netherlands has doubtless concentrated more on the role of cognitive styles in human-computer interaction than anyone else, although some of the team's work is rather speculative, such as the article by Van der Veer, Tauber, Waern and Van Muylijk (1985). Using Moran's (1981b) work as a starting point, the authors distinguish between various levels of interaction with the computer and discuss the potential influence of the user's characteristics on each. Other work is more empirical. Van der Veer and Beishuizen (1986), for example, have attempted to show the importance of the distinction proposed by Pask between serialist and holistic styles of learning (which are also known as **learning by operation** and **learning by comprehension**), a distinction that was shown to be important only in some of the tasks used. De Leeuw (1983) concentrated more on the distinction between field dependence and independence. He has shown that so-called field independent users can handle more easily the transfer from one computer system to another. The study by Egly and Westcourt (1982) in the United States showed that this same property also affected the way in which subjects used a database.

These results suggest that cognitive styles are a determining factor in human-computer interaction. This conclusion nevertheless shares the same ambiguities that characterize research into cognitive styles. Each study is generally based on a single classification only, so that it is difficult to know whether this classification is more important than the others, or even in what way it is different from them. Furthermore, the number of tasks considered is generally very limited. As we have mentioned, an individual's cognitive style is supposed to be a stable characteristic which is demonstrated in a relatively constant way in a variety of tasks. Some studies cited in Robertson's (1985) general review of the field show that this is not, however, always the case. Hudson (1968) suggests that the same subject may adopt convergent or divergent styles of thought almost on demand and an experiment by Webster and Walker (1981) showed that the style of thought adopted by subjects was also influenced by the nature of the test items with which the subjects were presented. If various cognitive styles are manifested in various tasks, it then becomes very difficult to distinguish styles from cognitive skills or even from strategies adopted by the subjects.

In sum, the concept of cognitive style is still surrounded by too great a degree of uncertainty, in our view, for it to be of any real use in the design of interfaces better adapted to individual users.

Occupations

The classifications described in the preceding sections claim to be based on fundamental psychological differences between individuals. Ergonomics research reflects a more pragmatic approach to the question of individual differences, focusing on problems which are concerned primarily with the nature of the jobs occupied by individuals and their level of skill. Borgman (1984) mentions that a recent report of the U.S. National Research Council on human-computer interaction lists no fewer than 17 different classifications based on degree of expertise with computers, and nine other classifications based on jobs.

Although Borgman's comment reflects the diversity of current research, it is not entirely fair. Rather than attempting to classify the various types of user of computer systems, most of the research cited in the report deals with only one target group of users. As far as jobs are concerned, the group of users most frequently studied is that of technicians, programmers and analysts. The predominant distinction is therefore that which separates people working in the computer field from those working outside it. This distinction reflects a difference in expertise, which will be covered in the next section, as much as it does a difference in jobs. Apart from those employed in computers, most attention has been focused on those working in the fields of administration and management.

Schoichet (1981, in Nickerson, 1986) distinguishes between three job categories. Taking the nature of the tasks performed as his starting point, he attempts to specify the characteristics which workstations should have if they are adapted to each of these categories. Thus, one type of workstation, the **functional** type, would be primarily for office clerks, assistants and secretaries. Such workstations should be designed in such a way as to facilitate data entry, text storage and retrieval, which are the tasks characteristic of such jobs. A different type of workstation would be required by professionals who make intensive use of computers, such as engineers, stockbrokers, loan officers, etc. According to Schoichet, workstations for these professionals should include, among other things, a high resolution screen and substantial storage capacity as well as being linked to a local area communications network. This last requirement also applies to the third type of workstation, that for managers, since statistics show that 75 percent of some managers' time is spent on communications functions.

It is difficult to specify the other properties which workstations for managers should have, since this group of users is quite heterogeneous. Senior managers thus spend much more time than lower-level managers on planning functions. Rockart and Treacy (1982, in Nickerson, 1986) use the term "executive information support" to designate the workstations required by senior managers. In addition to accessing data regarding the company and the market, such workstations should also, according to the authors, include systems to facilitate the analysis of such data and decision-making.

The nature of the tasks to be performed is not the only characteristic which distinguishes occupations. Eason, Damodaran and Stewart (1975; Damodaran, 1981; Eason, 1981; Stewart, 1981) note that the capacity and willingness to adapt to new technologies also varies greatly from one group of users to another. Managers generally have a choice as to whether to use technological tools or not, which is not the case for office staff who often have to adapt or lose their jobs. The latter thus have to learn the techniques they need for their work while managers, on the other hand, can delegate complex tasks to subordinates, who also act as intermediaries between them and computer systems. This solution is not always ideal, since, as Eason and his colleagues mention, managers can ignore some of the systems' capabilities and thus deprive themselves of such capabilities. Here again we encounter the problem of expertise.

Computer skills

The problem of computer skills has been present in a variety of guises since the computer was invented. In the early days, computers were used only by experts. The question of skills thus concerned only the learning of skills associated directly with computer science. With declining costs and the advent of minicomputers, the user population expanded to include various groups of professionals. Although they were not specialists in computers, these new users had sufficient general training to allow them to familiarize themselves, albeit not without some difficulty, with the basic techniques. Today, microcomputers are within the reach of the general population and as a result the focus has shifted to individuals with no training in computers. Such people are called **novices**.

Learning. The first questions to be asked with regard to novices concern learning. What must novices know in order to be able to use computer systems efficiently and how should they be taught? Although questions about learning go beyond the boundaries of traditional ergonomics, they have become very hard to avoid, given the complexity of computer technologies.

Some authors, including Anderson (1980), are of the opinion that users, in order to derive maximum benefit from computing tools, should have some knowledge of programming. This view raises the problem of the most suitable language for initiating users to programming. It is common knowledge that programmers have strong preferences as far as languages are concerned and it therefore comes as no surprise that these preferences are reflected in the debates surrounding the teaching of programming. Tagg (1974) reviews the various languages proposed and concludes that none of them is ideally suited for teaching purposes. Du Boulay, Oshea and Monk (1981) propose that programming be learned with the aid of a simple system of notation, while other authors tend towards computerizing the teaching of such commonly used programming languages as LISP and PASCAL.

Programming is an extremely complex cognitive activity. Nevertheless, some researchers claim that, in the future, all users of computer technologies will have some skills in this area. This belief is based on the fact that several educational programs already include courses in programming, while computer technologies have barely begun to be incorporated into various spheres of activity. In light of the expanding presence of such technologies, this educational trend can only increase and the result will be that future users will no longer be ignorant on the subject. This view is, clearly, not shared by all researchers. James (1981) believes that, on the contrary, the increase will be in the number of users with no prior knowledge. The complexity of computer systems is already so great that not all users can be expected to devote the effort necessary to familiarize themselves with the technology. James (1981) even proposes that an interface be developed to "protect" the user from the complexities of the system he is using by translating the user's requests into the system's language. The learning required in such a case would be limited to knowledge of the interface, which would itself be designed for easy understanding.

The situation desired by James is unfortunately still far from fruition. Without necessarily having to understand programming, users must still learn to manipulate the functions of the applications software if they wish to use the system interface. Such learning, however, presents many problems, even when the software is relatively simple, such as word processing.

One source of difficulty stems from the analogies that subjects establish between the new situation and others with which they are already familiar. Thus, Douglas and Moran (1983) note that a number of text editing errors arise from superficial similarities with simple typing. The mental model that novices apply to the situation thus has to be replaced by a more appropriate one. The studies by Mack and his colleagues (Carroll and Mack, 1983; Lewis and Mack, 1982) highlight the advantages of learning based on an appropriate model in comparison with learning based on rote learning of commands. In addition to correcting novices' often mistaken conceptions, the model makes the structuring of new knowledge possible.

It appears, nevertheless, that the advantage of learning by model is limited to tasks which require a certain element of reasoning. The findings of Borgman (1984) and Halasz (see Halasz and Moran, 1982, 1983) suggest that rote learning is as effective for simple, routine tasks. It is possible that the usefulness of models for learning is also related to the subjects' cognitive style, although to our knowledge no studies have yet been done on this question in the field of human-computer interaction.

The role of mental models is an area of research which is very active in cognitive science and many problems remain unsolved. Thus, the models presented to novices must be relatively simple if they are to be understood. They must, moreover, be sufficiently flexible to allow novices to progress and integrate their new knowledge as their skill increases. Another problem is that of the transfer of knowledge. The models used for learning one system must not constitute an obstacle to learning another. Some authors believe that no model can offer ideal qualities in all respects, and they consequently prefer to use the term "metaphor" rather than model.

Attitudes. The level of skill is not the only aspect which differentiates novices from experts. These two groups of users are also very different in their attitudes towards computers. Whereas the experts will try anything, or almost anything, to obtain the desired results, novices are sometimes hesitant to press a key. An anecdote may illustrate the problem. After taking delivery of her first computer, the chairman of a department at the University of Montreal was in a hurry to plug it in and insert the instruction diskette. The first message sent by the computer was illustrated with a picture of a bomb. It turned out that the computer was defective, but that did not prevent the purchaser from waiting several months before trying to use the machine again. This was, moreover, a computer advertised as being easy to operate and user-friendly. Some authors (Kennedy, 1975; Shneiderman, 1979) therefore stress that learning should not only serve to acquire knowledge but should also influence positively the attitudes of new users.

The interface. The experts' requirements with regard to a computer system are in almost permanent opposition to the needs of novices. This opposition stems from the fact that experts demand above all power from their systems, while novices need simplicity.

Thus, for example, menus are generally regarded as a more simple type of interface than commands. A study by Gilfoil (1982) shows that novices do in fact prefer to use menus. However, the same study shows that with experience, users rely increasingly on commands. With regard to command languages, Ledgard, Whiteside, Seymour and Singer (1980) compared two different systems, one based on a formal notation and the other closer to the users' natural language. The novices' performance was better with the second system.

Regardless of the system used, novices are more likely to make errors than experts. According to Shneiderman (1980b), the error messages generated by the computer should also be adapted to the user's level of skill. The cryptic messages intended for experts are all too often incomprehensible to novices. Shneiderman therefore proposes that messages intended for novices be more explanatory and provide the user with a list of options.

The needs of novices differ from the requirements of experts in several other respects too. Paxton and Turner (1984) provide a list of recommendations to facilitate the novice's task. Several of the messages proposed may be irksome for the expert user and merely waste time and space on the screen. This conflict would not be a problem if experts and novices always used different systems, although obviously this cannot be the case. In time, the novice user will become more "familiar" with the system, unless of course he remains no more than an occasional user. In an attempt to analyse this development, some authors have focused on intermediate categories of user between "novice" and "expert." Such classifications, however, consistute an essentially static approach which ill accords with the essentially dynamic nature of the changes that are at work in the user as he acquires greater skill.

In the preceding pages, we have reviewed briefly various ways in which users can be grouped. We have also seen some of the problems associated with the classifications, such as their static and ambiguous aspects. Because of these problems, those whose job it is to design new systems are often reluctant to base their decisions on these classifications. This reluctance was stated forcefully during a debate on the question which took place at the CHI'86 conference. For some of the participants in this discussion, computer systems should strive to meet the needs of each individual user rather than being based on necessarily too general characteristics of groups of individuals.

4.3 The individual user

One way of creating systems apapted to each user is therefore to create systems capable of adapting themselves to the user. To do this, the system must have knowledge or a model of the user. Rich (1983) proposes a distinction between user models that consist of three dimensions. The first ranges from canonical to individual models. The second contrasts models which develop implicitly, through interaction with the user, with models explicitly defined by the system designer, who then has to provide the system with parameters. The third dimension depends on the user characteristics incorporated into the model: some (long-term) models take only the user's stable characteristics into account, while others (short-term) are centred around the characteristics of the user which can change in the course of the interaction.

These three dimensions are not symmetrical with each other nor are they independent of the topics discussed in the various sections of this chapter. Thus a model of the typical user, for example, can only be canonical, by definition. It will probably be explicit and long-term. One example of this type of model is the ZOG program developed by Robertson, McCracken and Newell (1981) to facilitate human-computer communication. Explicit, long-term models are also suitable for representing groups of users. As far as individual users are concerned, only long-term models can be defined explicitly. A system can easily ask the user his age, educational level and other, relatively stable aspects of his personality. On the other hand, users find it much less acceptable to have their task constantly interrupted by requests to specify explicitly to the system the aim of the exercise at hand. Short-term models should therefore be implicit: the user's aims and knowledge must be inferred by the system.

Short-term models are precisely the ones which are most likely to facilitate human-computer interaction. Once it knows the user's aim, the system can not only detect erroneous methods being used, but also, in the case of correct methods, instruct him in the use of more efficient ones. This type of personalized instruction, often known as "embedded tutoring," differs from more traditional teaching methods in that it is provided at the moment when the user manifests a need for it.

Unfortunately, the problem of inferring goals and knowledge on the basis of actions is one of the most complex in cognitive science. As far as goals are concerned, Perrault and his colleagues (Cohen and Perreault, 1979) have developed a system which has proved moderately successful in inferring intentions hidden behind (or revealed through) the statements of an interlocutor. This research is based in part on the theories of Searle (1969) and Grice (1975), on language and on the seminal research by Newell and Simon (1972) on goal structures. Other relevant research on planning is by Sacerdoti (1977). With regard to the inference of knowledge, some research deals with knowledge which is specific to a given field. This type of research is amply illustrated in Sleeman and Brown's (1982) book on tutoring systems. Other work concentrates on specifying general methods by which knowledge can be inferred. The recent article by Doignon and Falmagne (1985) is an example of this type of research. Even if the research cited in this paragraph is basic and does not deal directly with human-computer interaction, work focusing on such applications is currently in progress in a number of institutions.

Chapter 5

The task

The task, the third component of the user-computer system, is omnipresent since, as we pointed out earlier, it is the raison d'etre of the interaction. It nevertheless seems to have remained the stepchild of ergonomics research. Of course, all empirical studies are carried out in the context of a specific task, but as a general rule more attention is paid to the characteristics of the command system and to the type of dialogue than to the task itself, as perceived by the user. The question is generally asked as follows: given task X and relatively well-defined user category Y, what are the characteristics of the system interface which promote efficient interaction? There is still little interest in what features distinguish one task from another from the standpoint of 1) the problem-solving structure and 2) ergonomic requirements. The question is nevertheless vital to an overall assessment of ergonomic results and models, particularly in the context of integrated systems where the user is required to perform several tasks simultaneously (Miyata and Norman, 1986).

The researchers face one major obstacle, however: no one has yet achieved a consensus on a taxonomy of tasks to provide a basis for analysis, even though that has been recognized for some time as a major step, as Melton and Briggs pointed out in 1960:

> "a taxonomy of tasks (...) is a necessary condition for systematic and confident application of engineering psychology principles to decisions regarding alternative system designs involving different human functions."
>
> (Melton and Briggs, 1960, p. 90)

Companion and Corso (1982) reviewed research that had attempted to classify tasks in the general context of human-computer interaction. They then evaluated the five major taxonomies surveyed in terms of 11 efficiency criteria which they had formulated themselves. None of the taxonomies met more than eight criteria and consequently Companion and Corso concluded that there is still no classification system capable of combining in a useful tool the data on human performance which has been gathered to date.

In the absence of such an analytical tool, we have proposed the three dimensions illustrated in the model presented in Chapter 2 (see figure 2): variations in the task domain, in its complexity and in the structure of the operating language. These three sources of variation cannot constitute a true classification system, since they are not a unidimensional continuum or even three mutually independent dimensions.

This distinction simply seemed to us to offer a convenient method of presenting the various facets according to which the problem of the task can be tackled. In the following pages, we will examine in detail only the dimension of the task domain; the problems of complexity and of the structure of operating langugages will not be covered in separate sections, but will be discussed within each of the domains.

The task domain

Word processing, consulting databases, electronic mail, graphics manipulation, process control, programming and so on are all domains which require appropriate forms of interaction. This classification according to the nature of the task is not truly scientific, but reflects in general the way in which users perceive these applications and it is, moreover, the way software is marketed. Clearly, the user's activities are duplicated to a considerable extent from one domain to another: for example, word processing, electronic mail and programming all include a substantial amount of input and manipulation of textual data. Each domain, however, requires certain special skills, which in turn give rise to specific problems.

In this section we will present some of the research which has been done in some of these task domains, stressing in particular the main tasks of the automated office: word processing, databases and electronic mail.

5.1 Word processing

Word processing or text editing is without doubt the task which has generated the largest amount of research, although this should come as no surprise in that it is by far the most widespread and widely used application. It is also the task with the largest pool of potential users, which is why it is of particular interest from the standpoint of variables that determine **ease of learning**. Basing himself on Moran (1981b), Douglas (1983) shows that full page display, the mnemonic command names, the use of insert as a default command (see also Poller and Garter, 1983), the use of a mouse and function keys are features which reduce the conceptual complexity of the editor and facilitate learning. These findings confirm among others many studies of the mnemonicity of commands as a key factor in learning (Barnard et al., 1982; Black and Moran, 1982; John, Rosenbloom and Newell, 1985; Mack, 1985; Scapin, 1981; Rosenbery, 1982, 1983).

Considerable attention has also been paid to **learning methods**: Lewis and Mack (1982), after analysing the protocols of subjects in independent learning situations, conclude that self-teaching cannot be a productive method in view of the complexity of the task and the virtual impossibility of providing documentation which anticipates all possible situations. Observing that novice users often rely on analogies derived from their knowledge of the typewriter, Douglas and Moran (1983) demonstrate that there are both advantages and pitfalls associated with this learning by analogy, but that it is possible to predict and sometimes to forestall the errors that often stem from this method (see also Allwood, Eliasson and Wikstrom, 1986; Galambos, Wikler and Black, 1983; Halasz and Moran, 1982). Ross and Moran (1983) show the efficiency of learning that is based, not on an analogy with the typewriter, but on the analogy between the problem at hand and problems previously solved by the user.

Carroll and Mack (1983) assert that the novice user is not a passive learner and that, on the contrary, he proceeds actively by experimenting and producing himself inferences regarding the material to be learned. They therefore suggest that the user be provided with a conceptual learning framework, as well as an environment which promotes exploration and learning by doing. These recommendations regarding conceptual and active learning were subsequently adopted by Bourdeau (1986), which confirms their relative validity.

Other research has focused on variables linked to the **characteristics of the user**. Egan et al. (1982) show that speed of learning depends among other things on reading skills, spatial memorization capacity and factors which vary with age. Egan et al. continued (1983) their work, however, by pointing out the interaction between the characteristics of the users and of the editor. For example, as we mentioned earlier, the effect of age seems to decrease significantly when learning a full page editor rather than a line editor. Tyler, Roth and Post (1982) and Rosson (1983) show that learning is only very rarely complete and that even those who can be described as expert users truly master all the functions only in exceptional cases. This is partly why several studies, including those of Foss, Rosson and Smith (1982), Hold, Boehm-Davis and Schultz (1986) and Magers (1983) have concentrated on the problem of **documentation** and help systems to encourage and maintain learning, as well as on error messages.

Attention has shifted more recently to the **transferability** of learning from one word processing software package to another. Karat, Boyes, Weisberger and Schafer (1986) show that while the learning of general concepts can be transferred, proactive interference occurs with regard to the syntax of functions, which varies substantially from one software package to another. Ziegler, Hoppe and Fahnrich (1986) have studied the transfer from word processing to graphics manipulation and found that transfer is better from learning text to graphics than vice versa. It is nevertheless questionable whether this conclusion can be generalized without further verification.

Several hundred types of word processing or text editing software package have been launched on the market in the past few years. It would consequently have seemed normal for a **comparative evaluation** to have been made of the various products that would go farther than the comparative lists of features that are so numerous in popular magazines and "user's guides." The obvious usefulness of such an exercise notwithstanding, only Roberts and Moran (1982, 1983; Roberts, 1979) have undertaken a systematic comparison of the various editors. Roberts (1979) first analyses the editors and compiles a list of 212 subtasks which can be performed on a document. Based on this taxonomy, she calculates the functionality of each editor by computing the ratio: "tasks which can be performed by the editor"/"set of 212 possible tasks." No editor achieved a functionality higher than 77 percent and the average was 54 percent. Roberts uses 53 basic subtasks, common to nine of the editors in her study, which are subsequently used to compare the editors from the point of view of learning speed, execution time and percentage of time spent correcting errors. There were marked differences between the editors, with learning speed varying from one to four times, execution speed varying by a factor of 2.5 and errors by a factor of 5.

Borenstein (1985) uses Roberts and Moran's benchmark set of tests to evaluate two other editors and revalidate their test, with similar results. However, he points out certain problems inherent in these tests and criticizes in particular the definition of an expert user, based entirely on cumulative use time and not on any directly measurable skill.

The work of Roberts and Moran was epoch-making in the field because it produced the first tangible, standardized tool for the concrete evaluation of word processing software. The test nevertheless has its limitations. First, the tool can be used only after the fact, once use of an editor is sufficiently widespread for expert users to be found. It cannot therefore be used in the design phase, except to anticipate functionality. Secondly, the taxonomy of "possible tasks" is not absolute and reflects, as Borenstein (1985) points out, the environment in which the people who developed it are working. Thirdly, the validity of the test tends to diminish with time, as the capacities of editors evolve. The method therefore requires periodic updating, which is an inevitable problem in the context of a constantly evolving technology. It is, moreover, remarkable that to date no one has evaluated more recent editors in the same way.

Another interesting approach consists of analysing the **behavior of writers** in a real situation, as Embley and Nagy (1982) have done. Hoffman, Ziessler and Seifert (1986) examine the problems which occur in filling in forms. Note also the studies by Allen (1981, 1982a) and Gould (1980, 1981) on the composition and revision of manuscripts. It is clear that many users now tend to compose their texts directly at the keyboard and that this activity merits more attention.

The most interesting research from the theoretical point of view, however (and doubtless the most promising from a practical point of view in the longer term) are those which attempt to formulate **more general models** of the user's representation of the task. Since Oren (1974), who proposes a purely mathematical model of text editing, several researchers have focused on the structure of knowledge and the user's problem solving procedures (in other words, the goal structure or production rules). Folley and Williges (1982), for example, demonstrate that the patterns of command use reflect differing models of the task in the novice and the expert.

Robertson and Black (1983) affirm that the behavior of text editing can and should be considered not as a linear succession of keystrokes, but as a planned behavior governed by a goal hierarchy. Based on Schank and Abelson's (1977) theory, they develop and test a model of editing as a sequence of graded plans, a sequence reflected in the grouping of keystrokes by the user.

Card, Moran and Newell (1983) concur in a major study resulting from several years of work (Card et al., 1976; Card et al., 1980a, 1980b; Card, 1982), several chapters of which are devoted to text editing. Rather than attempt to describe here a study which cannot be reduced to a few lines, we will simply point out that they develop several models of the task from the point of view of the goal structure, models which vary significantly depending on the thrust of the analysis. Card et al. nevertheless describe text editing as a "routine cognitive skill," thereby implying that the behavior studied is that of expert subjects. They also devote very little attention to errors. The quality of the analyses of this Xerox PARC group is unanimously recognized in the research community, but the difficulty of this "computational" approach (see Chapter 1) is such that very few others have dared to extrapolate it to other tasks or to study more complex editing situations.

More recently, however, other researchers have pursued this approach, representing the task as a production system. This is the case with Polson and Kieras (1985), who base their approach on their theory of complexity (Kieras and Polson, 1985) to create a simulation which succeeds in predicting learning difficulty and performance time. Ziegler, Vosson and Hoppe (1986) also use an analysis from the point of view of production rules to predict transfers of learning.

What can we conclude from all this work on text processing or editing? The first lesson is doubtless that this task (or rather this set of tasks and subtasks) is a complex cognitive phenomenon which can be understood only if several dimensions are taken into account. This is what makes word processing interesting, but it must be admitted that it is a laborious task to produce results which are both generalizable and directly applicable.

In this sense, word processing is representative of the other task domains, even though research in it is significantly farther advanced. There are numerous studies on very specific problems, but their results are either difficult to generalize or highly ephemeral because they are very closely tied to the system studied. This is true, for example, of studies such as that by Schneider, Nudelman and Hirsh-Pasek (1982), which deal with line numbering strategies in text editors, for which the results are outdated for virtually all recent editors, or that of Gould, Lewis and Barnes (1985) on the effects of cursor speed, which does not apply to direct manipulation interfaces.

At the other extreme there are excellent analyses such as those by Card et al. (1983), which are more generalizable but of less immediate applicability. For all that, however, there is no call for pessimism: the research surveyed is all useful in designing and evaluating specific software and the more theoretical studies are partly responsible for recent and future developments in intelligent programs to help the user.

5.2 The databases

From the point of view of ergonomic research, a distinction can be drawn between the domain of databases and that of text editing in a variety of dimensions, four of which seem to us fundamental.

The first dimension involves the extent of the domain of information concerned. While types of word processing software differ in several respects, they nevertheless remain a fairly homogeneous domain. Although earlier there were fairly marked variations depending on the application (editor or formatters, program editors, etc), today's software is tending towards ever greater uniformity and the performance of functions which are basically similar. Research, as we saw earlier, can therefore concentrate on certain problems such as command syntax, types of interaction and the construction of a general model of the task. The domain of databases is much more heterogeneous and we cannot talk of databases as just another "application." In fact, the domain of information storage and retrieval, which is hidden behind the term "database," constitutes an entire domain of research in informatics. By way of evidence of this, while there exists an interest group of the Association for Computing Machinery (ACM) for the domain of human-computer interaction (SIGCHI), the domain of databases has another interest group all to itself, the Special Interest Group on Information Retrieval (SIGIR). From the point of view of concrete applications, the range is very broad: large public databases such as videotex, bibliographical databases, corporate databases, inventory management systems, decision-making support systems, as well as electronic bulletin boards, diaries and calendars, file management systems and so on. As we will see farther on, it is still possible to find general problems, but a very high proportion of problems are context-specific, which makes any synthesis extremely difficult.

The second dimension which distinguishes databases and word processing has to do with the nature of the task itself. Word processing consists of entering and manipulating information by following certain relatively invariable rules: it is thus possible to define, as Roberts (1979) did, the word processing universe with a considerable degree of precision. This also enables the user to master this application fairly quickly and allows the task to become a routine cognitive skill in fairly short order. The task of word processing is, moreover, relatively independent of the information content being handled and the size of the file, except for certain easily identifiable functions involving locating.

The problems of databases lie elsewhere and are significantly more complex: it is no longer a question of inputting the information, but of organizing it into a structure which facilitates selective handling and retrieval. This structure will, furthermore, be intimately linked to the semantics of the information stored. A specialized database containing well defined classes of properties (i.e., the title, author and year of publication, in a bibliographical database) is not organized in the same way as a videotex database designed for public use and containing highly diversified content. In addition, and as opposed to word processing, the problems of data organization and retrieval vary depending on the size of the base. For example, a system with hierarchical menus can be envisaged for a small or medium-sized database, but becomes inefficient for a very large database, particularly one with heterogeneous content.

This difference in the very nature of the two types of task gives rise to a third source of variation, one which is linked to criteria of ergonomic efficiency. As we pointed out, word processing quickly becomes a routine activity; the work of the researchers is aimed primarily at speeding up the attainment of this level of skill and minimizing performance time by reducing errors. The same may hold true for databases, but not in all cases. It is doubtless possible to attain this level of skill in a client or inventory file that is used on a daily basis and in which all the information is classified in a number of well defined fields. It is, however, quite different in a large, heterogeneous file, the content of which is less well defined in advance, as would be the case for example in a thematic search of a bibliographic database. In such cases (and we know whereof we speak...), the task is a genuine problem solving activity and the first performance criterion to be met is **the efficiency of the query.** To use the language of sign detection theory, we have to evaluate the proportion of "misses" (information desired but not retrieved) and "false alarms" (information retrieved but not desired) and develop a system which reduces these two categories in favor of "hits" (information desired and retrieved). A word processing system may be poorly designed and include a number of irritating aspects, but eventually the user will generally achieve his goal: a database, on the other hand, can be absolutely unusable.

The fourth dimension which differentiates word processing and databases has to do with **command syntax and embedding.** In word processing, a typical subtask like the replacement of one word by another requires only a short sequence of commands, and the revision of a text can be perceived as a series of such sequences which are relatively independent of each other. The task becomes a reiteration of subtasks with little grading. Similarly, the syntax and grammar of the command language are very limited, to the extent that the number of "phrases" permitted is finite and countable.

In a complex database, a query may take the form of a fairly long sequence of chained and mutually dependent commands, which can attain the complexity of a short program and require a good deal of planning on the part of the user. The form of the query is therefore much more variable, and research into query languages lies at the heart of ergonomic research into databases. This is why, in the context of this study, we shall restrict ourselves to a discussion of certain studies which have been carried out to develop and evaluate query languages. To do this, we will draw on two overview articles which in our view accurately reflect the state of the field, by Reisner (1981) and especially the article by Jarke and Vassiliou (1985).

Research on query languages

In her review of research on query languages, Reisner (1981) establishes several interesting distinctions which have lost none of their relevance with the passage of several years. In particular, Reisner attempts to clarify the concept of ease of use of query languages. She distinguishes first between three dimensions which differentiate query languages. They can vary by syntactical form, with some languages being linear, as in many keyword systems, and others bidimensional, as in Query by Example systems (Zloof, 1983). A distinction can also be drawn between procedural and non-procedural languages, depending on whether the query consists of specifying a sequence of stages to arrive at the desired subset or simply specifying the characteristics of the subset. The languages can also vary according to the apparent structuration of the data stored, in other words, the form in which the data presented to the user is organized. This structure may be **relational** (in the form of a table), **hierarchical** (in the form of an inverse tree) or a **network** (in the form of a graph which can be heterarchical), Reisner then enumerates the tasks used to measure ease of use as well as the criteria for the ergonomic evaluation of languages. The tasks selected are writing, reading and interpretation of a query, memorization of the structure of data and generation of multiple queries for a single problem. In terms of ergonomic criteria, Reisner mentions some of the general criteria cited in Chapter 2 (speed of learning, speed of performance and retention), but adds others such as efficiency of the search from the point of view of the information retrieved (which we discussed above) and the speed of relearning, in other words, the ease with which an occasional user relearns elements of the language he has forgotten.

Reisner then gives several examples showing how these tasks and these criteria have been used to measure quantitatively ease of use and to compare several query languages. Thus, Greeblatt and Waxman (1978; in Reisner, 1981) demonstrate that in a relational database, the use of a bidimensional syntax (query by example) facilitates learning and makes queries faster and more precise than relying on a linear syntax. Welty and Stemple (1979; in Reisner, 1981) focus on procedurality and show that procedural and non-procedural languages are equivalent for simple queries, but that procedural languages are superior for complex requests. As for the effects of the apparent structuring of data, Lochovsky and Tsichrizis (1977; in Reisner, 1981) compare relational, network and hierarchical databases and find that, for users who are relatively unfamiliar with the domain of the content of the database, the relational structure more frequently generates correct queries than the network base, which in turn is superior to the hierarchical base. These differences vanish to a large extent once the users become more familiar with the content of the database.

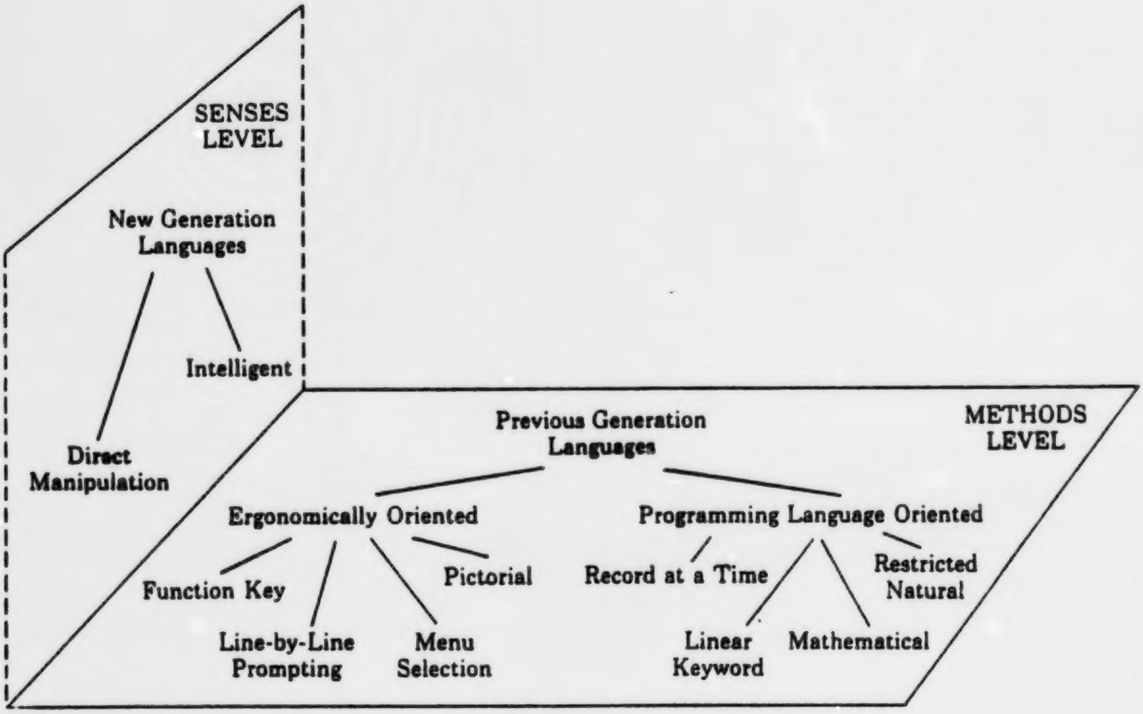
Caution is nevertheless indicated in interpreting these findings. As Reisner emphasizes on several occasions, the generality of such experimental results is not measurable and it is essential to take into account the conditions of the experiment and especially the characteristics of the bases tested.

Of all the research papers we have encountered, the article by Jarke and Vassiliou (1985) constitutes the most remarkable overview of the domain of query languages. Jarke and Vassiliou set themselves the goal of developing a methodology for evaluating query languages from the point of view of functionality and user-friendliness, a methodology that would take the type of user into account. While we shall not describe the scheme they propose in detail, it is based on three taxonomies which deserve attention: the classification of languages, of users and of evaluation criteria.

In terms of languages, Jarke and Vassiliou (see Vassiliou and Jarke, 1982) classify languages in two dimensions (see Figure 4): the first is that of query methods and allows the languages to be organized as they existed in the early '80s. According to the authors, the languages developed historically by following two trends: languages oriented towards ergonomic characteristics or stemming from concerns that have more to do with informatics. Jarke and Vassiliou describe the characteristics of four query methods for each trend.

Figure 4

Classification of query languages (Jarke and Vassiliou, 1985)



The second dimension, at right angles to the first, is that of the sensory modalities used and allows recently developed languages to be organized. Note for example the use of direct manipulation interfaces (Shneiderman, 1986a), "spatial" databases, which enable the user to explore the databases like a three-dimensional environment (Herot, 1984), and multimodal bases (Bolt, 1984; Schmandt, 1985; Schmandt and Hulteen, 1982). Jarke and Vassiliou also enumerate the main "intelligent" systems, which enable the user to ask questions about the structure of the data and help him to formulate his query (Tou, Williams, Fikes, Henderson and Malone, 1982) or which interact in natural language and acquire new knowledge by experience (Schank and Slade, 1984; in Jarke and Vassiliou, 1985).

In the context of users, Jarke and Vassiliou suggest a classification based on abilities in interaction (which would be a combination of the user's familiarity with programming concepts and his frequency of use of the language in question) and abilities in the domain (which are determined by his knowledge of the structure and content of the database and the range of queries that the user is called upon to execute). This typology has the benefit of being more appropriate to the domain of databases than other more general classifications and of recognizing the difference between ability in interaction and with content, a distinction already identified by Streitz (1986).

Jarke and Vassiliou, like Reisner (1981), also compile a list of criteria for ergonomic evaluation, but in their case the typology is considerably more advanced, as Figure 5 shows.

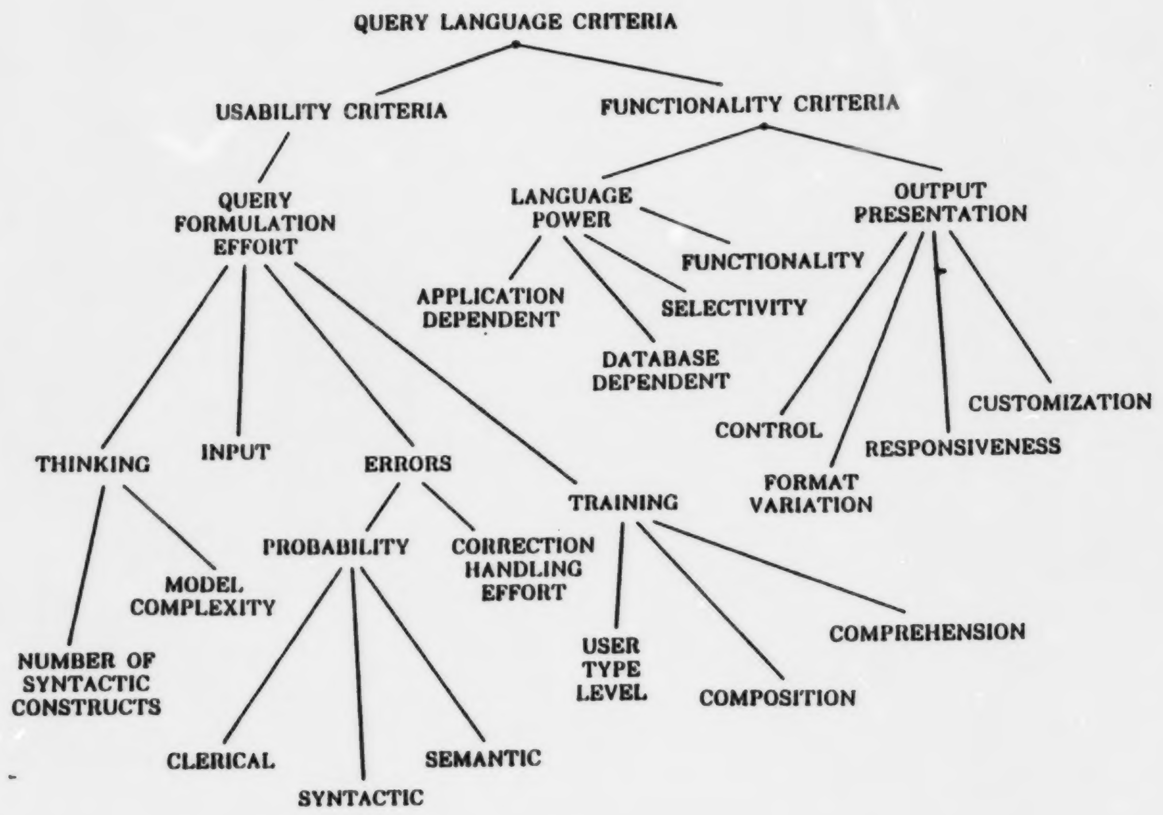


Figure 5

Classification of ergonomic criteria (Jarke and Vassiliou, 1985)

For Jarke and Vassiliou, ergonomic efficiency is based on two dimensions whose relationship to each other is at times unstable, and which the user would evaluate in terms of the **cost/benefit** ratio: user-friendliness and functionality. **User-friendliness** represents the cost or the effort required to manipulate the system. In the case of a database, it is a question of evaluating the effort required to query the system: "thinking," length of the input, errors produced and learning. **Functionality** represents the potential of the language and determines the benefit that can be anticipated for the user from the point of view of the value of the information retrieved. There are two aspects to consider: the **power** of the language, which results from **the variety of possible queries** and the **selectivity of requests**, i.e., the capability of retrieving the information desired and only that. Power is also a factor of the relative independence of the language in relation to the range of possible applications and to the various types of apparent structure of the data. The second aspect is linked to the result of a query and to the user's power to control the quality and form of the output.

In the case of databases, functionality is probably a more important criterion than for word processing, since the power required varies widely depending on the application and the user's needs.

Jarke and Vassiliou continue with an examination of these three typologies of language, of users and of ergonomic criteria to formulate their evaluation methodology. Although the actual forecasting value of this method is open to question, their study seems to us to reflect the current state of our knowledge of query languages.

Before we conclude this section on databases, we would like to draw the reader's attention to recent attempts to understand more fully the advantages and limitations of the use of natural or pseudo-natural language (Dumais and Landauer, 1982; Koch, 1986; Ogden and Brooks, 1983; Small, 1983; Turner, Jarke, Stohr and Vassiliou, 1984), since it is not always clear that the use of natural language is optimal in the context of information retrieval (Hayes, 1985). Mention should also be made of another avenue of research which is not directly concerned with the problem of query languages, but rather with the **exploration** of data. It is a well known fact that one of the current limitations of computer systems is the "bottleneck" of the screen in terms of information flow. The problem is particularly evident in databases, where access to information is considerably reduced by this limited visualization space. This is why several researchers have approached this problem by imagining systems that would enable them to "navigate" within the data. We would cite among others the work of Herot (1984) and the MIT group (Bolt, 1984), McDonald's (1982) multimedia approach, Jones' (1986) "memory extension" system and Furnas' (1985, 1986) "fisheye" view.

5.3 Electronic mail

Of all the possible applications of automated systems in the workplace, computer-mediated interpersonal communication is one of those with the greatest potential impact on corporate communications. Electronic mail is only one of the forms of computer-mediated communication, which also includes file sharing for joint work, teleconferencing, terminal to terminal real-time communication and screen-sharing systems (Bannon, 1986; for a more general overview, see Hiltz, 1983; Hiltz and Turoff, 1978; Kerr and Hiltz, 1982).

The primary function of an electronic mail or messaging system is to enable messages, generally text, to be sent and received. The messages are stored in a "mailbox" and retrieved on demand by the addressee. Messages are sent simultaneously to one or more addressees, who may be linked to the same computer or alternatively share a messaging system linking computers located around the world, as in the case of networks such as ARPANET. The advantages are of course the instantaneous transmission of mail to one or more users and especially the capability of contacting someone who is physically absent at the time of the transmission, which avoids the problems associated with the use of the telephone, such as the interruption of work in progress and in particular missed telephone contacts, in which the parties try in vain to reach each other leaving endless messages.

Electronic mail systems generally include basic word processing functions accompanied by commands for transmission, receipt and addressing. In appearance, the task thus comes close to word processing mixed with specific file management commands (the messages) and does not seem to pose any particular problems. Some studies of users and researchers nevertheless show that factors outside the system itself are most often the cause of user-friendliness problems.

After evaluating the most important ergonomic characteristics amongst researchers who had worked on nine mail systems, Turoff, Hiltz and Kerr (1982) conclude that habitual factors (type and mnemonicity of commands, etc.) have only a secondary effect. The use of messaging systems is actually unique in that the user often has great latitude with regard to the frequency with which he uses the system. The users' skills and occupational situation vary widely, which assigns prime importance to motivational factors. In some contexts, where use of the system is intensive and necessary to the operation of the organization, the users succumb to environmental pressures to reply to their messages within a fairly short time and a reply that takes more than a day would be a sign of "bad manners" (Bannon, 1986). In many other cases, the proportion of assiduous users has not yet attained a "critical mass" and use of the system is still based on the user's goodwill; in this case, ease of use becomes of prime importance.

Hersh (1982) identifies another interesting phenomenon in his study of the use of a management mail system. Of managers who are provided with the services of a secretary, 50 percent do not use the system themselves, but send and receive messages through the secretary. Often, the secretary herself drafts the messages and filters those intended for her boss, who receives only the written copies: this type of use is very much in line with more traditional means. In addition, managers who learn to use the system generally do so by trial and error, even if they are offered a three-hour course (see also Hiltz and Kerr, 1986). While there is nothing inherently negative about this reorientation of the use of electronic mail, it does reflect the importance of motivational factors in the use of such systems. For managers who have a secretary and are unfamiliar with the use of a keyboard, the perceived cost of use is not worth the benefit of direct use, and it may well be that it would take systems with a speech-recognition capability to satisfy them.

The only true ergonomic problem that is specific to the task of messaging has to do with the phenomenon of "electronic junk" and poses the problem of the selective retrieval of information. When several hundred users are linked, one of the current messaging strategies is for the user to rely on "mailing lists" which reflect his interests. The sender of a message thus has to choose his group or groups of addressees. Unfortunately, many users tend to be unselective in their distribution, with the result that one may find oneself one fine morning with several dozen, if not several hundred messages waiting to be read, with no way of telling which are relevant. This is the problem of electronic junk, which can become counter-productive, as a number of researchers have pointed out (Denning, 1982; Hiltz and Turoff, 1985; Palme, 1984; in Malone, Grant and Turbak, 1986).

Although several possible solutions have been suggested (Hiltz and Turoff, 1985; Stallman, 1983; in Malone et al., 1986), we will concentrate here on the prototype developed by Malone et al. (1986). Using a direct manipulation interface, the proposed system consists of supplying the user with a variety of semi-structured templates for drafting and reading his messages. The aim is akin to storing messages in the form of a relational database. The user can filter or file his messages depending on the values selected in certain basic domains. When he sends a message, the user can send a copy to a special file known as an "information lens," which automatically distributes the message to all users who have specified the same values on the template. One of the advantages of the prototypes of Malone et al. is that the development of the templates is based on empirical analysis of the nature of the messages exchanged within the organization, which promotes an information flow appropriate to the organization as well as saving the user time in consulting his messages.

Conclusion

We have covered here only three of the most common tasks in the domain of office automation. Research has been done on several other task domains, including programming, graphics in its limitless variety of forms and process control (especially in the field of air traffic control). Mention could also be made of all the applications related to computer-aided learning, design and manufacturing, decision-making support systems, etc. It should be noted, furthermore, that there is now a tendency for task domains to blend within integrated software, which makes it possible to combine text, graphics and databases with which the user can perform a variety of tasks (see on this subject Cypher, 1986 and Miyata and Norman, 1986).

It is clearly impossible to survey all these studies, which alone account for publications numbering in the hundreds. The purpose of this chapter was rather to show how research into human-computer interaction is not only dependent on the type of communication between the user and the system, but also on the task or tasks in question. As we have seen, each task poses certain problems of cognitive ergonomics that are specific to itself. This is why a distinction is increasingly being drawn between the semantics of interaction and those of the task (Streitz, 1986; Norman, 1984). As Norman points out (1984, p. 60), system design requires three sorts of knowledge: the first is that of design, programming and technology; the second is knowledge of the human being and mental models; the third is that of the task to be performed. Designers generally have a good knowledge of design, but neither of the user nor of the task. Psychologists have a good knowledge of the user, but not of design or of the task. While the user is familiar with the task but not the two components. Intelligent system design therefore demands the formation of teams that contain all three types of expertise.

Chapter 6

Recommended research agenda

In the preceding chapters we attempted to sketch as general an outline as possible of the various types of research done in the cognitive ergonomics of computer systems. We hope that this review of the literature will have convinced the reader that cognitive ergonomics is a research field that is young, vigorous and expanding rapidly. It was also our intention to demonstrate that cognitive ergonomics suffers from the defects associated with its youth, namely a certain lack of precision in the definition of the subject of its study, a proliferation of methods and a degree of anarchy with regard to the problems considered. We therefore believe that it is important in this concluding chapter to propose a number of research avenues which, in our view, have the potential to lead to an improved understanding of human-computer interaction and thereby contribute to the development of more user-friendly interfaces.

This chapter thus differs from its predecessors in several respects. First, it is more selective. We will touch on only a few specific problems, and even here we will limit ourselves to an indication of their importance for future research. Secondly, this chapter is more subjective than previous ones, since the topics discussed have been selected from several possibilities. Our choices are not necessarily incompatible with those of other researchers in the field, such as Shneiderman (1986b), but our aim is not to reflect faithfully the suggestions found in the literature. One final comment is in order. The purpose of this chapter is to suggest several research topics and not to develop research projects. It is therefore more a question of presenting a number of interesting problem areas rather than giving an accurate description of experimental methods and plans.

6.1 Direct manipulation interfaces

Direct manipulation interfaces were developed following the advent of techniques that made possible full screen display and addressing each point (or pixel) on the screen. With these techiques, it became easy to "manipulate" the "objects" displayed on the screen. There are already direct manipulation interfaces for a variety of tasks, ranging from word processing to computer-aided design. Some computer systems, the best known of which is doubtless the Macintosh, have been designed especially for direct manipulation. However, since these developments are relatively recent, little is yet known about the advantages and disadvantages of direct manipulation as a way of interacting with the computer.

The advantages most frequently mentioned are the greater ease of learning and use of direct manipulation interfaces in comparison with command interfaces, which require the user to master an abstract and often arbitrary syntax. Even if these advantages are intuitively plausible, there is still a shortage of quantitative data to support its existence. A number of experiments have been conducted, but few have compared direct manipulation with other types of interface while keeping the task software identical in both cases. For example, word processing software programs which allow direct manipulation vary in several respects from command software, so that the structure of the task to be performed can vary with the type of interaction. It is difficult to draw clear conclusions in such cases with regard to the type of interaction alone.

Another problem concerns the possibility of generalizing the results obtained from a specific task to other tasks. Some tasks, such as computer-aided design, appear to lend themselves quite naturally to direct manipulation, since they are activities which are essentially oriented towards objects of a spatial nature. But what about tasks of a verbal and semantic nature, such as the retrieval of information from a database? And what about such essentially abstract tasks as programming? Do direct manipulation interfaces facilitate the learning and performance of tasks? These questions have received very little attention to date. Questions regarding the ease of learning and use of systems meet only some of the criteria for evaluating interfaces. What about the other criteria listed in Chapter 2, such as the response times of the systems? Do direct manipulation interfaces encourage shorter response times or do they, in contrast, detract from the development of skill? All these questions require urgent consideration in view of the current status of interface technologies.

It is of course possible to counter that these questions are too closely linked to our current technologies and that technical progress will have made them obsolete even before answers have been found. There can be no doubt that presently available interfaces will one day be outdated. In this context, Gaines and Shaw (1986) distinguish several stages in the development of interface techniques, ranging from a phase of exploring the possibilities inherent in a new technology to a phase of mass production and reproduction and a subsequent phase of knowledge of principles, and so on. To some extent, direct manipulation interfaces have already gone beyond the exploratory phase, although it cannot be categorically stated that they have entered the mass production phase, since most existing systems still have not yet been equipped with them. On the other hand, some of the research in progress (such as the MIT projects described in Chapter 3) are intended to refine the potential of direct manipulation, which has not yet been fully developed. We can therefore conclude that direct manipulation interfaces have more than a dozen years of useful life remaining, which leaves sufficient time for research in this area to produce useful results. It is perhaps appropriate to bear in mind that technologies do not always reach the market as soon after they are invented: the famous mouse, which experienced its market breakthrough in the early 1980s, was patented before 1970.

We believe, moreover, that the future of direct manipulation interfaces raises important theoretical questions. Although such interfaces are well defined physically, this is much less true in cognitive terms, since the objects represented and the manipulations possible may be different. It is plausible to think that the success of such interfaces will depend on the faithfulness with which the images on the screen (icons) and their manipulation will correspond to the users' mental representations and to the ways in which they can affect these representations. We still know very little about the mental representations in question, for example, in the design of a car body, even though, as we have already mentioned, this activity is essentially spatial in nature. What happens with the spatial representation of abstract information, such as the date of Abraham Lincoln's birth? What icon should we use to represent this type of information? And in what "form" should we represent the database containing information about the presidents of the United States, so that users can "steer" through it without getting lost? The question whether humans can represent in image form data which is not intrinsically so is still being debated vigorously in cognitive psychology. We still know virtually nothing about how users handle it and the extent to which this skill varies in the population. These problems, which are related to the question of users' mental models, are thus of theoretical as well as practical importance.

Before we turn to the question of mental models, it may be useful to point out once more the importance of direct manipulation interfaces. The cognitive ergonomics of computer systems began with the advent of interactive computers. In ergonomic terms, the great leap forward in interactive computers stems from their ability to respond almost immediately to the user's requests. This technology has thus led us to view work on the computer as an interaction similar to interactions between human partners. It is therefore not surprising that the ideal interface has often been described as the one which permits a dialogue in natural language between user and machine. Direct manipulation interfaces have the potential to change radically this view of work on the computer. It is no longer a question of **interacting** in a verbal and abstract way with an invisible partner, but rather of **acting** concretely on visible objects. The user is no longer an interlocutor but rather an actor. Such a change of perspective will inevitably have major consequences, not only on research in cognitive ergonomics but also on the concept of work in general, even though it is difficult at present to predict the exact nature of the impact.

6.2 Mental models

Our second recommendation is intended to encourage research into users' mental models. The concept of the mental model is derived from one of the most fundamental postulates of cognitive psychology, according to which human beings act and reason as a function of their mental representation of the world around them. Human beings are in constant interaction with their environment, and a high proportion of human cognition consists of abstracting from this interaction a representation of the world and the rules that govern it. Human beings thus construct for themselves a model of how the world operates, and they act and anticipate the consequences of their actions in terms of this model. The reliance on mental models is omnipresent, whether to interpret the intentions of an interlocutor or to determine why an automobile has broken down.

Semantic analysis of mental models is a relatively recent development and one which until now has focused primarily on an understanding of mathematical or physical systems, such as fluid mechanics (Hayes, 1979, in Gentner and Stevens, 1983), electricity (Gentner and Gentner, 1983), thermodynamics (Wiser and Carey, 1983) and movement (McCloskey, 1983). Some researchers have nevertheless concentrated on the mental model which people construct of the machines they use. In particular, several recent research projects have examined the use of pocket calculators (Halasz and Moran, 1982, 1983; Young, 1981, 1983)) and show clearly that the users generally have a very incomplete and often inadequate model of how their calculators work. For example, users frequently refuse to use certain functions such as the memory, because they are not sure how they work, or develop superstitious beliefs such as pressing the "clear" key several times before each new calculation. Several studies (Halasz and Moran, 1983; Kieras and Bovair, 1984) show, moreover, that an adequate mental model of how a device works facilitates the learning and retention of commands and procedures, generates more efficient strategies and facilitates recovery in the event of error.

When applied to the domain of human-computer interaction, the concept of the mental model is defined as the user's representation of the operation of the system with which he is interacting. In the case of a computer, the development of an adequate mental model is all the more important in that users generally have a scant basis on which to construct one. In the case of these physical devices, there is in fact generally a correlation between structure and function: in other words, how the device works can be deduced in part from an examination of its physical characteristics. Furthermore, interaction with this type of device provides indications from a variety of sources: movement of the parts, spatial relationships between the components, sounds and so on.

Computer systems are far more "opaque" in the sense that it is difficult for the user to understand what is happening "behind the screen" (Brown, 1986). Even though the system does emit a number of physical signals, such as the sound of the diskette moving, the console lights or the response time, these signals are of little use without a complex mental model. Consequently, and although we still know little about the subject, it is probable that the mental models of users of computer systems present boundaries which are just as important as those described by Norman (1983b).

1) The mental models are incomplete.

2) The users' ability to execute their models (i.e., to make them work mentally) is very limited.

3) The mental models are unstable: users have a tendency to forget certain details of their model.

4) The mental models have no clearly defined boundaries: users tend to confuse models of similar software programs.

5) There is nothing scientific about the mental models: users persist with their superstitious behavior, even when they know it is pointless, if it requires little physical effort and saves them mental effort.

6) The mental models tend to be parsimonious: users prefer to perform a few more actions, if this saves them from more complex mental planning. This is particularly true if the user can repeat the same set of simple actions, in which case he will prefer a long

series of actions to the inclusion of more complex commands in his model, even is his behavior is inefficient.

It therefore seems to us of prime importance to understand the models developed spontaneously by users and above all to analyse how to promote the construction of more satisfactory mental models. Some researchers have already proposed some general principles to solve this problem, such as conceptual learning and the use of metaphors, but little work has been done on testing these solutions. The construction of an adequate mental model nevertheless offers several advantages, which we will summarize in conclusion.

a) Learning based on a mental model is much more easily transferable.

b) A complete mental model allows the user to understand the situation and to imagine solutions in the event of breakdown or error.

c) Since his understanding of the system is greater, the user can constantly adjust his choice of the optimal strategy.

d) A user with a satisfactory mental model is more easily encouraged to explore the system's full potential and to experiment with new strategies.

e) In terms of motivation, a user with a rich mental model has much better control of the interaction, and this constitutes an important positive reinforcement.

6.3 Intelligent interfaces

Like it or not, computers are complex tools. While there are certainly ways to make them somewhat easier to use, we believe that there are limits beyond which simplification is paid for in loss of power, generality, efficiency and flexibility. One solution to this problem is to teach the user to learn about his tool. The converse solution, which consists of teaching the machine to "know" its user, is more in keeping with the aims of cognitive ergonomics.

We saw in Chapter 4 how taxonomies of users are sometimes contradictory, often imprecise and always unsatisfactory, since the taxonomies are set while the users evolve. It is therefore impossible to design a system which is adapted in advance to its users and which remains so over time. On the contrary, we must design systems capable of recognizing the user's individuality, noticing the changes occurring in him over time and adapting to these changes. This ideal places work on artificial intelligence at the very centre of the development of cognitive ergonomics.

Systems must first of all be capable of recognizing the user's level of expertise. The problems of a novice user will not be those of an expert, whose problems will be different again from those of an occasional user. Research into artificial intelligence with the aim of developing methods by which the user's level of knowledge can be inferred are therefore essential. In the absence of such methods, it is pointless to hope to create tutorial systems and computerized consultants, since such systems will know neither what to teach, nor to whom nor when to teach it.

The systems' recognition and inference capacities must not be limited to the user's level of knowledge alone. They must also extend to the user's aims and intentions. Recognizing intentions is a difficult undertaking, since, in contrast to users' knowledge, which remains relatively stable over the short term, the user's aims change in the course of the interaction as they are achieved. In addition to knowing the user, the system must know the domain which the user is attempting to perform a task.

We know that humans are to some extent capable of recognizing the intentions of others and that good teachers are able to recognize their students' difficulties, but we still know little about the indicators and strategies used in this. To be effective, research into artificial intelligence must therefore be accompanied by research in cognitive psychology. In view of the scope of the problems involved, there should be no expectation that the work will bear fruit immediately, but we are convinced that in the longer term, the most important developments in cognitive ergonomics will result from the development of intelligent interfaces.

6.4 The organizational interface

Throughout this study, we have focused on the cognitive aspects of the user-computer interaction, in an attempt to show the relevance and current status of research in the new field of the cognitive ergonomics of computer systems. Our final recommendation will go somewhat beyond the strict user-task system, in order to underline the importance of the organizational context within which the interaction takes place and which is as much a determining factor as the strictly cognitive factors discussed up to now.

Our personal experience of the effective use of office automation in the workplace indicates that the quality of the interaction is often severely hampered by constraints related to the actual work situation. Let us consider two simple examples. In one of the organizations with which we have had the opportunity to become familiar, a word processing system was introduced into a small department in the following way: as soon as the single machine arrived and was installed, a secretary was offered an intensive course on it lasting several days. When she returned to the department, it was assumed that her mastery of the equipment was now perfect and the secretary from then on was expected to perform at normal if not higher than normal levels. After more than two years of use, we observed significant gaps in this person's knowledge of the operation of the system. The reasons for this seem quite clear. First, it must be stressed that a learning process lasting for a few days, even if well done, is generally insufficient to allow internalized learning of all the commands. The most complex commands are often no more than touched on, while the time is spent on simpler and more frequently used ones. A modern word processor has considerably more than a hundred commands and procedures and the simultaneous learning of all these concepts overloads the average learning capacity, especially since the learning is most often procedural (i.e., from the point of view of the sequence of commands to be executed) rather than conceptual (i.e., providing a mental model of how the system functions).

The user is, moreover, expected to operate at full efficiency as soon as the learning has been completed, and as a result no allowance is made for further training sessions, since the user has presumably seen the full range of commands. The result of all this is that the user, confronted with often pressing demands from his superiors, selects the most urgent one and learns to get by with a limited number of simple functions which he knows well, without ever having the time to explore the more complex commands that he has practised on only a few occasions. The user no longer has time to consult the manual and no one is there to point out his shortcomings. It is worth noting in passing that the secretary in our first example was soon assigned the task of initiating her colleagues into the operation of the word processor...

Our second example is similar: a secretary using one of the most common word processors saw that her immediate superior had acquired a recent microcomputer with a direct manipulation interface. She asked for permission to use this new marvel on occasion, not only for word processing but also to perform graphics applications and to assemble a database. To enable her to do this, he gave her a few minutes' lesson and the manuals (which of course were in English). The result was entirely predictable: with this new machine, on which learning apparently takes place by exploration, the secretary wasted a great deal of time fumbling around in order to achieve rather mediocre results, since little of her previous knowledge was transferable to the new and much more versatile machine. The secretary consequently developed a highly negative attitude towards the new machine, even though in theory it was much more user-friendly than its predecessor designed exclusively for word processing.

There are many other examples which could be cited in which the problems of cognitive ergonomics are in large measure due to organizational factors. The reader will recall among others the restricted use which managers often make of electronic mail systems (see Chapter 5). The two examples we have described here deal with a learning situation, but learning should not be viewed as the only initial stage in using a system. It is vain to hope that everything can be learned at the outset; a considerable portion of the development of real expertise must occur during use. Furthermore, the systems' rapid evolution makes all learning relatively ephemeral and the technology demands constant retraining. This is why two of the preceding recommendations deal with the acquisition of adequate mental models and the development of intelligent interfaces capable, among other things, of guiding the user after the initial learning phase. This problem of the organizational interface poses the problem of the systems' "usability," which we discussed in Chapter 2. The design of future systems will take into account the actual conditions in which the technology is introduced, so that its full potential can be exploited. Designers must not only take the users into account, but also the conditions under which they work. This is why cognitive ergonomics must sometimes come out of the laboratories to analyse the interaction as it actually occurs in the "real" world. The three avenues of research which are most important are, in our view: a) a cognitive analysis of the use made of technology when it has been installed for a certain length of time; b) an analysis of the structure of the users' activities, so that a better understanding can be achieved of how technology fits into this context; c) an analysis of the organizational conditions which promote the exchange of knowledge between peers (see on this subject Bannon, 1986). As we pointed out in the initial chapters of this study, the frontiers of cognitive ergonomics are still not clearly defined. This apparent gap may turn into an advantage if researchers use it to open up the focus of their research.

Periodicals consulted

Annual Review of Information Science and Technology (USA)

Applied Ergonomics (UK)

Behaviour and Information Technology (UK)

Ergonomics (UK)

Ergonomics Abstracts (UK)

Hewlett Packard Journal

Human Factors (USA)

Human-Computer Interaction (USA)

IBM Systems Journal

IEEE Transactions on Software Engineering (USA)

International Journal of Man-Machine Studies (USA)

Le Travail humain (human work) (France)

Proceedings of the Computer-Human Interaction Conferences (USA)

Bibliography

ABELSON, P.H. (1982) The revolution in Computers and Electronics. Science, 215, 751-753.

ALDEN, D.G., DANIELS, R.W., and KANARCK, A.F. (1972) Keyboard design and operation: a review of the major issues. Human Factors, 14, 275-293.

ALLEN, R.B. (1981) Composition and editing of text. Ergonomics, 24, 611.

ALLEN, R.B. (1982a) Patterns of manuscript revisions. Behaviour and Information Technology, 1(2), 177-184.

ALLEN, R.B. (1982b) Cognitive Factors in Human Interaction with Computers. Behaviour and Information Technology, 1, (3), 257-278.

ALLWOOD, C.M., ELIASSON, M., and WIKSTROM, T. (1986) Analogical thinking in Human Computer Interaction. Proceedings of the 3rd European Conference on Cognitive Ergonomics, (Paris, September 15 to 19, 1986), INRIA, 107-114.

ANDERSON, B. (1980) Programming in the home of the future. International Journal of Man-Machine Studies, 12, 341-365.

ANDERSON, J.R. (1985) Cognitive psychology and its implications. New York: Freeman, 2nd edition.

BANNON, L.J. (1986) Computer-Mediated Communication. In D.A. NORMAN and S.W. DRAPER (eds.), User centered system design. Hillsdale, New Jersey: Lawrence Earlbaum Associates, 433-452.

BARNARD, P.J., HAMMOND, N., MacLEAN, A. and MORTON, J. (1982) Learning and Remembering Interactive Commands. Behaviour and Information Technology, 1, (4), 347-358.

BARNARD, P.J., HAMMOND, N.V., MORTON, H. and LONG, J.B. (1981), Consistency and Compatibility in Human-Computer Dialogue. International Journal of Man-Machine Studies, 15, 87-134.

BEWLEY, W.L., ROBERTS, T.L., SCHROIT, D. and VERPLANK, W.L. (1983) Human Factors Testing in the Design of Xerox's 8010 "Star" Office Workstation. Proc. CHI'83 Human Factor in Computing Systems, (Boston, December 12-15), ACM, New York, 72-77.

BIRNBAUM, J.S. (1982) Computers: A survey of trends and limitations. Science, 215, 760-765.

BLACK, J.B. and MORAN, T.P. (1982) Learning and Remembering Command Names. Proc. CHI'82, Human Factors in Computing Systems, (Gaithersburg, Maryland, March 15-17), ACM, New York, 8-11.

BOLT, R.A. (1984) The Human Interface, where People and Computers Meet, Belmont, California: Lifetime Learning Publications.

BORENSTEIN, N.S. (1985) The Evaluation of Text Editors: A Critical Review of the Roberts and Moran Methodology Based on New Experiments. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18, 1985), ACM, New York, 99-106.

BORGMAN, C.L. (1984) Psychological Research in Human-Computer Interaction. In M.E. WILLIAMS (Ed.) Annual Review of Information Science and Technology, 9, 33-64.

BOURDEAU, J. (1986) Apprentissage conceptuel du traitement de texte à l'aide de jeu, simulation et tutorat (conceptual learning of word processing with the help of games, simulation and tutoring). Unpublished PhD. Thesis, Department of Educational Technology, University of Montreal.

BROWN, J.S. (1986) From cognitive to social ergonomics and beyond. In Norman, D. and Draper, S.W. (eds.) User centered system design. Hillsdale, Laurence Earlbaum Associates.

BURCH, J.L. (1984) Computers: The non-technological (Human) Factors: A recommended Reading list on Computer Ergonomics & User Friendly Design, The Report Store, MA., 94 p.

BUXTON, W. (1986) Human interface design and the handicapped user. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, p. 291.

CARD, S.K. (1982) User perceptual mechanisms in the search of computer command menus. Proc. CHI'82 Human Factor in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 190-196.

CARD, S.K. (1984) Human Factors and the Intelligent Interface, Technical report #84.2-01, Xerox Palo Alto Research Center, Palo Alto, CA.

CARD, S.K., MORAN, T.P. and NEWELL, A. (1976) The manuscript editing task: A routine cognitive skill, Technical report #55 L 76-8, Xerox Palo Alto Research Center, Palo Alto, CA.

CARD, S.K., MORAN, T.P. and NEWELL, A. (1980a) Computer text-editing: An information processing analysis of a routine cognitive skill. Cognitive Psychology, 12, 32-74.

CARD, S.K., MORAN, T.P. and NEWELL, A. (1980b) The keystroke-level model for user performance time with interactive systems. Communication of the ACM, 23, (7), 396-410.

CARD, S.K., MORAN, T.P. and NEWELL, A. (1983) The Psychology of Human-Computer Interaction. Hillsdale, New Jersey: Lawrence Earlbaum Associates.

CARD, S.K. and NORMAN, D. (1986) Resolved: Interface Design Doesn't Matter - A Debate. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, p.89.

CARROLL, J.M. and MACK, R.L. (1983) Learning to use a word processor: by doing, by thinking, and by knowing. In J. THOMAS and M. SCHNEIDER (eds.) Human Factors in Computer Systems. Norwood, New Jersey: Ablex Publishing. 13-51.

CARROLL, J.M. and KAY, D.S. (1985) Prompting, Feedback and Error Correction in the Design of a Scenario Machine. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18), ACM, New York, 149-153.

CHAPANIS, A., PARRISH, R.M., OCHSMAN, R.B. and WEEKS, G.D. (1977) Studies in interactive communication: II. The effects of four communication modes on the linguistic performance of teams during cooperative problem solving. Human Factors, 19, 101-126.

CHIN, D.N. (1986) User Modeling in UC, the UNIX Consultant. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, 24-28.

COLE, W.G. (1986) Medical Cognitive Graphics. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17, 1986), ACM, New York, 91-95.

COHEN, P. and PERREAULT, C.R. (1979) Elements of a plan-based theory of speech acts. Cognitive Sciences, 3, (3), 177-212.

COMPANION, M.A. and CORSO, G.M. (1982) Task taxonomies: a general review and evaluation. International Journal of Man-Machine Studies, 17, 459-472.

COOPER, W.E. (1982) Cognitive aspects of skilled typewriting. New York: Springer-Verlag.

CYPHER, A. (1986) The structure of user's activities. In D.A. NORMAN and S.W. DRAPER (eds.) User centered system design. Hillsdale, New Jersey: Lawrence Earlbaum Associates. 243-264.

DAMODARAN, L. (1981) The role of user support. In B. SHACKEL (ed.) Man-Computer Interaction: Human factors aspacts of computers and people, Rockville, Maryland: Sijthoff and Noordhoff.

DEATHERAGE, G.H. (1972) Auditory and other sensory forms of presentation. In H.P. VAN COTT and R.G., KINKADE, (eds.) Human Engineering Guide to Equipment Design. Revised edition, Washington, D.C.,: U.S. Government Printing Office, 123-160.

DE LEEVW, L. (1983) Teaching problem solving: An ATI study of the effects of leaching algorithmic and heuristic solution methods. Instructional Science, 12, 1-48.

DE LEON, L., HARRIS, W.G. and EVENS, M. (1983) Is There Really Trouble With Unix? Proc. CHI'83, Human Factors in Computing Systems, (Boston, December 12-15), ACM, New York, 125-134.

DE MONTMOLLIN, M. (1986) L'ergonomie (ergonomics). La Découverte, Paris.

DENNING, T.J. (1982) Electronic Junk. Communication of the ACM, 163-165.

DESMARAIS, M., GIROUX, L. and LAROCHELLE, S. (1986) Model calculatoire du processus cognitive d'inférence des plans (calculational model of the cognitive inference of plans) (summary). Annales de l'ACFAS, Montreal, May 1986.

DEVOE, D.B. (1967) Alternatives to handprinting in the manual entry of data. IEEE Transactions on Human Factors in Electronics, HFE-8, 21-31.

DIEBOLD, J. (1969) Man and the Computer. New York: Praeger.

DOIGNON, J.P. and FALMAGNE, J.C. (1985) Spaces for the assessment of knowledge, International Journal of Man-Machine Studies, 23, 175-196.

DOUGLAS, S. (1983) What features make a text editor easy to learn. Standford University and Xerox Palo Alto Research Center, Palo Alto, CA. p. 13.

DOUGLAS, S.A. and MORAN, T.P. (1983) Learning text editor semantics by analogy. Proc. CHI'83 Human Factor in Computing Systems, (Boston, December 12 to 15), ACM, New York, 207-211.

DREYFUS, H.L. (1979) What Computers can't do: A Critique of Artificial Reason, New York: Harper and Row.

DU BOULAY B., O'SHEA, T. and MONK, J. (1981) The black box inside the glass box: presenting computing concepts to novices. International Journal of Man-Machine Studies, 14, 237-249.

DUMAIS, S.T. and LANDAUER, T.K. (1982) Psychological investigations of Natural Terminology for Command & Query Languages. In A. BADRE and B. SHNEIDERMAN (eds.), Directions in Human-Computer Interaction, Norwood, New Jersey: Ablex Publishers.

EASON, K.D. (1980) Dialogue design implications of task allocation between man and computer, Ergonomics, 23, 881-891.

EASON, K.D. (1981) A task-tool analysis of manager computer-interaction. In B. SHACKEL (ed.) *Man-Computer Interaction: Human factors aspects of computers and people*, Rockville, Maryland: Sijthoff and Noordhoff, 289-307.

EASON, K.D. (1984) Towards the Experimental Study of Usability. *Behaviour and Information Technology*, *3* (2), 133-143.

EASON, K.D., DAMODARAN, L. and STEWART, T.F.M. (1975) Interface problems in Man-Computer Interaction. In E. MUMFORD and H. SACKMAN (eds.), *Human choice and computers*, Amsterdam: North Holland, 91-105.

EDMONDS, E. (1982) The Man-Computer Interface: a note on Concepts and Design. *International Journal of Man-Machine Studies*, 16, 213-236.

EGAN, D., BOWERS, C. and GOMEZ, L. (1982) Learner characteristics that predict success in using a text-editor tutorial. *Proc. CHI'82 Human Factor in Computing Systems*, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 337-340.

EGAN, D. and GOMEZ, L. (1984) Assaying, isolating and accommodating individual differences in learning a complex skill. In R.F. DILLON (ed.) *Individual differences in cognition: Volume 2*, New York: Academic Press.

EGAN, D., GOMEZ, L., WHEELER, E.A., SHARMA, D.K. and GRUCHACZ, A.M. (1983) How Interface Design Determines Who Has Difficulty Learning to Use a Text Editor. *Proc. CHI'83 Human Factors in Computing Systems*, (Boston, December 12 to 15, 1983), ACM, New York, 176-181.

EGLY, D.J. and WESCOURT, K.T. (1982) Cognitive style, categorization, and vocational effects on performance of relational data-base users. *Bulletin of the Association for Computing Machinery (ACM). Special Interest Group on the Social and Behavioral Science of Computing (SIGSOC)*, ACM, New York, 337-340.

EMBLEY, D. and NAGY, G. (1982) Can We Expect to Improve Text Editing Performance? *Proc. CHI'82 Human Factors in Computing Systems*, (Gaithersburg, Maryland, March 15 to 17, 1982), ACM, New York, 152-156.

ERICSSON, K.A. and SIMON, H.A. (1985) *Protocol Analysis, Verbal Reports as Data*. Cambridge, Massachesetts: MIT Press.

EVANS, C. (1980) *Les géants minuscules*. Interéditions, Points.

FEIGENBAUM, E. and McCORDUCK, P. (1983) *The fifth generation: artificial intelligence and Japan's computer challenge to the world*. Reading, Mass: Addison-Wesley.

FISCHER, G., LEMKE, A. and SCHWAB, T. (1985) Knowledge-based Help Systems. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18), ACM, New York, 161-167.

FOLLEY, L.J. and WILLIGES, R.C. (1982) User Models of Text Editing Command Languages. Proc. CHI'82, Human Factors in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 326-331.

FOSS, D., ROSSON, M.B. and SMITH, P. (1982) Reducing Manual Labor: An experimental analysis of learning aids for a text editor. Proc. CHI'82 Human Factor in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 332-336.

FURNAS, G.W. (1985) Experience with an adaptive indexing scheme. Proc. CHI'85, Human Factors in Computing Systems, (San Francisco, April 14 to 18), ACM, New York, 131-136.

FURNAS, G.W. (1986) Generalized fisheye view. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, 16-23.

GAINES, B.R. and SHAW, M.L.G. (1983) Dialog engineering. In M.E. SIME and M.J. COOMBS (eds.): Designing for Human-Computer Communication, London: Academic Press.

GAINES, B.R. and SHAW, M.L.G. (1986) From timesharing to the sixth generation: the development of human-computer interaction. Part 1. International Journal of Man-Machine Studies, 24, 1-27.

GALITZ, W.O. (1981) Handbook of screen for format design. Wellesley, MA.: Q.E.D., Information Science, Inc.

GALAMBOS, J.A., WIKLER, E.S. and BLACK, J.B. (1983) How You Tell Your Computer What You Mean: Ostension in Interactive Systems. Proc. CHI'83 Human Factors in Computing Systems, (Boston, December 12 to 15, 1983), ACM, New York, 182-185.

GENTNER, D. and GENTNER, D.R., (1983) Flowing water or teeming crowds: Mental models of electricity. In D. GENTNER, and A.L. STEVENS (eds.), Mental Models, Hillsdale, New Jersey: Lawrence Earlbaum Associates.

GENTNER, D. and STEVENS, A.L. (1983) Mental Models, Hillsdale, New Jersey: Lawrence Earlbaum Associates.

GILFOIL, D.M. (1982) Warming up to computers: A study of cognitive and affective interaction over time. Proc. CHI'82 Human Factor in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 245-250.

GIROUX, L. (1983) La dimension cognitive de la bureautique (the cognitive dimension of office automation) Face à la nouvelle technologie. Actes du quatrième congrès de l'Association de la Recherche en Communication du Québec (proceedings of the 4th congress of the Quebec association for communications research), Montreal, 3 p.

GIROUX, L. (1986) Communication et cognition: l'interaction usager-ordinateur (communication and cognition: the user-computer interaction), Communication et Information, 3(1), 170-180.

GIROUX, L. (1986) L'ergonomie cognitive des systèmes informatiques (the cognitive ergonomics of computer systems), L'intelligence artificielle et l'informatisation du travail : Actes de l'atelier (proceedings of the colloquium on "Artificial Intelligence and Office Automation") (Laval, April 29), Laval: CWARC.

GIROUX, L. and BELLEAU, R. (1986) What's on the Menu: the Influence of Menu Content on the Selection Process. Behaviour and Information Technology.

GIROUX, L., BERGERON, G., and LAMARCHE, J.-P. (publication pending) Organisation sémantique des menus dans les banques de données (semantic organization of database menus). Le Travail Humain.

GODFREY, D., MADDEN, J., PARKHILL, D.F. and OUIMET, A. (1980) Gutenberg Two. Victoria: Press Porcepic.

GOMEZ, L.M., EGAN, D.E., WHEELER, E.A., SHARMA, D.K. and GRUCHACZ, A.M. (1983) How interface design determines who has difficulty learning to use a text editor. Proc. CHI'83 Human Factor in Computing Systems, (Boston, December 12 to 15), ACM, New York, 176-184.

GOOD, M. (1985) The use of logging data in the design of new text editor. Proc. CHI'85 Human Factors in computing Systems, (San Francisco, April 14 to 18, 1985), ACM, New York, 93-98.

GOULD, J.D. (1980) Experiments on compsing letters: some facts, some myths, some observations. In L.W. GREGG and E.R., STEINBERG (eds.), Cognitive process in writing, Hillsdale, New Jersey: Lawrence Earlbaum Associates.

GOULD, J.D. (1981) Convoying letters with computer based text-editors. Human Factors, 23, 593.

GOULD, J.D., LEWIS, C. and BARNES, V. (1985) Effects of cursor speed on text-editing. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18, 1985), ACM, New York, 7-10.

GRANDJEAN, E. (1969) Précis d'ergonomie, Dunod, Paris.

GRICE, H.P. (1975) Logic and conversation. In P. COLES and J.L. MORGAN (eds.) Studies in Syntax, Vol. III. New York: Academic Press.

GRUDIN, J. and BARNARD, P. (1985) When does an abbreviation become a word? And related questions. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18, 1985), ACM, New York, 121-125.

GUILFORD, J.P. (1967) The nature of human intelligence. New York: McGraw-Hill.

HALASZ, F.G. and MORAN, T.P. (1982) Analogy considered harmful. In Proc. CHI'82 Human Factor in Computing Systems (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 383-386.

HALASZ, F.G. and MORAN, T.P. (1983) Mental models and problem solving using a calculator. Proc. CHI'83 Human Factor in Computing Systems, (Boston, December 12 to 15), ACM, New York, 212-216.

HARRIS, D.H. (1986) Fundamentals of Human Factors Criteria for User Interface Design. Tutorial 9, CHI'86 Human Factor in Computing Systems, (Boston, April 13 to 17), ACM, New York, 212-216.

HARTSON, H.R. (1985) Advances in Human-Computer Interaction. Vol. 1, Norwood, New Jersey: Ablex Publishing Corporation.

HARVEY, O.J., HUNT, D. and SCHRODER, H. (1961) Conceptual systems and Personality Organization. New York: Wiley.

HAYES, P.J. (1985) The utility of natural language interface. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18, 1985), ACM, New York, 121-125.

HECKEL, P. (1984), The Elements of Friendly Software Design. Warner Books, New York, 205 p.

HEMENWAY, K. (1982), Psychological Issues in the Use of Icons in Command Menus. Proc. CHI'82, Human Factors in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 20-23.

HENDERSON, A. (1986) The Trillium User Interface Design Environment. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17) ACM, New York, 221-227.

HEROT, C.F., (1984) Graphical user interface. In Y. VASSILIOU (ed.), Human Factors and Interactive Computer System. Proc. of the New York Symposium on User Interfaces, (New York, May 26 to 28, 1982), Norwood, New Jersey: Ablex Publishers Corporation. 83-104.

HERSH, H. (1982) Electronic Mail Usage Analysis. Proc. CHI'82 Human Factor in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 278-280.

HILTZ, S.R. (1983) On line communities. Norwood, New Jersey: Ablex Publishers Corporation.

HILTZ, S.R. and KERR, E.B. (1986) Learning modes and subsequent use of computer-mediated communication systems. Proc. CHI'86 Human Factors in Comuting Systems, (Boston, April 13, to 17, 1986), ACM, New York, 149-155.

HILTZ, S.R. and TUROFF, M. (1978) The Network nation: human communication via computer. Mass: Addison-Wesley.

HILTZ, S.R. and TUROFF, M. (1985) Structuring computer-mediated communication systems to avoid information overload. Communication of the ACM, 28 (7), 680-689.

HINTON, G.E. and ANDERSON, J.A. (1981) Parallel models of associative memory. Hillsdale, New Jersey: Lawrence Earlbaum Associates.

HIX, D. and HARTSON, R.H. (1986) An Interactive Environment for Dialogue Development: Its Design, Use, and Evaluation - or - Is Aide Useful? Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, 228-234.

HOFFMANN, J., ZIESSLER, M. and SEIFERT, R. (1986) Psychological Methods for Assembling Procedures in Text Management Systems. In F. KLIX and H. WANDKE (eds.). Proc. of the First Network Seminar of the International Union of Psychological Science on Man Computer Interaction Research: MACINTER - I, (Berlin RDA, October 16 to 19), Amsterdam: North-Holland, 117-123.

HOLLNAGEL, E. and WOODS, D.D. (1983) Cognitive Systems Engineering: New Wine in New Bottles. International Journal of Man-Machine Studies, 18, 583-600.

HOLT, R.W., BOEHM-DAVIS, D.A. and SCHULTZ, A.C. (1986) The effects of structured, multi-level documentation. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17, 1986), ACM, New York, 122-128.

HOWARD, D.V. (1983) Cognitive psychology: Memory, language and thought. New York: Macmillan Publishing Co.

HUDSON, L. (1968) Frames of mind. Harmondsworth: Penguin Books.

INGENITO, C. and GIROUX, L. (in preparation) Ergonomie cognitive et logiciels de bases de données (cognitive ergonomics and database software). M.A. Thesis, University of Montreal.

JAMES, E.B. (1981) The user interface: how we may compute. In M.J. COOMBS and J.L. ALTY (eds.) Computing skills and the user interface. New York: Academic Press.

JARKE, M. and VASSILIOU, Y. (1985) A framework for choosing a Database Query Language. Computing Surveys, 17, 313-340.

JOHN, B.N., ROSENBLOOM, P.S. and NEWELL, A. (1985) A Theory of Stimulus-Response Compatibility applied to Human-Computer Interaction. Proc. CHI'85, Human Factors in Computing Systems, (San Francisco, April 14 to 18), ACM, New York, 213-219.

JOHNSON-LAIRD, P.N. (1980) Mental Models in Cognitive Science, 4, 71-115.

JONES, W.P. (1986) The Memory Extender Personal Filing System. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, 298-305.

KAGAN, J., ROSMAN, B.L., DAY, D., ALBERT, J. and PHILLIPS, N. (1964) Information processing in the child: significance of analytic and reflective attitudes. Psychological Monographs, 78.

KANTOWITZ, B.H. and SORKIN, R.D. (1983) Human Factors Understanding People-System Relationships. New York: John Wiley and Sons.

KARAT, J., BOYES, L., WEISGERBER, S. and SCHAFER, C. (1986) Transfer Between Word Processing Systems. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17, 1986), ACM, New York, 67-71.

KENNEDY, T.C.S. (1975) Some behavioral factors affecting the training of naive users of an interactive computer system. International Journal of Man-Machine Studies, 7, 817-834.

KERR, E.B. and HILTZ, S.R. (1982) Computer mediated communication systems. New York: Academic Press.

KIERAS, D. and BOVAIR, S. (1984) The role of a mental model in learning to operate a device. Cognitive Science, 8, 255-274.

KIERAS, D. and POLSON, P.G., (1985) An approach to the formal analysis of user complexity. International Journal of Man-Machine Studies, 22, (4), 365-394.

KINDEAD, R. (1975) Typing speed, keying rates, and optimal keyboard layouts. Proc. of the 19th Annual meeting of the Human Factors Society.

KOCH, U. (1986) User Requirements in Natural Language Communication with Database Systems, In F. KLIX et H. WANDKE (eds.) Proc. of the First Network Seminar of the International Union of Psychological Science on Man Computer Interaction Research: MACINTER - I, (Berlin RDA, October 16 to 19), Amsterdam: North-Holland, 285-294.

LACHMAN, R., LACHMAN, J. and BUTTERFIELD, E.C. (1979) Cognitive Psychology and Information Processing: An Introduction, Hillsdale, New Jersey, Lawrence Earlbaum Associates Publishers.

LANDAUER, T.K., GALOTTI, K.M. and HARTWELL, S. (1983) Natural command names and initial learning: A study of text-editing terms, Communication of the ACM, 26, (7), 495-503.

LANDAUER, T.K. and NACHBAR, D.W. (1985) Selection from alphabetic and numeric menu trees using a touch screen: breadth, depth and width. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18), ACM, New York, 73-78.

LAVILLE, A. (1976) L'ergonomie (ergonomics). Presses Universitaires de France, Paris.

LEDGARD, H.F., WHITESIDE, J.A., SEYMOUR, W. and SINGER, A. (1980) An experiment on human engineering of interactive software. IEEE Transactions on Software Engineering, SE-6, 6, 602-604.

LEE, S.K., BUXTON, W. and SMITH, K.C. (1985) A multi-touch three dimensional touch-sensitive tablet. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18) ACM, New York, 21-26.

LEWIS, C. and MACK, R. (1982) Learning to use a text processing system: Evidence from "thinking aloud" protocols. Proc. CHI'82 Human Factor in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 387-392.

LICKLIDER, J.C.R. (1960) Man-Computer Symbiosis. Institute of Radio Engineers Transactions on Human Factors Electronics, AFE-1, 4-11,

LINDE, L. (1986) The information-seeker's mental model of the content of a data base. Actes de la 3e conférence européenne d'ergonomie cognitive, (Paris, September 15-19, 1986), INRIA, 129-138.

LINDSAY, P.H. and NORMAN, D.A., (1977) Human Information Processing, An Introduction to Psychology, 2nd edition, New York: Academic Press.

LUCAS, R.W., CARD, W.I., KNILL-JONES, R.P., WATKINSON, G. and CREAN, G.P. (1976) Computer Interrogation of Patients. British Medical Journal, 2, 623.

MACK, R. (1985) Identifying and designing toward new user expectations in a prototype text-editor. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18, 1985), ACM, New York, 139-141.

MAGERS, C.S. (1983) An experimental evaluation of on-line HELP for non programmers. Proc. CHI'83, Human Factors in Computing Systems, (Boston, December 12 to 15), ACM, New York, 277-281.

MAGUIRE, M. (1982) An Evaluation of Published Recommendations on the Design of Man-Computer Dialogues. International Journal of Man-Machine Studies, 16, 237-261.

MALONE, T.W. (1980) What Makes Thinks Fun to Learn? A Study of Intrinsically Motivating Computer Games, Technical report #CIS-7, Xerox Palo Alto Research Center, Palo Alto, CA.

MALONE, T.W. (1981a) What Makes Computer Games Fun? Byte, 6, 258-277

MALONE, T.W. (1981b) Toward a Theory of Intrinsically Motivating Instruction. Cognitive Science, 5, publication pending.

MALONE, T.W. (1982) Heuristics for Designing Enjoyable User Interfaces: Lessons from Computer Games. Proc. CHI'82, Human Factors in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 63-68.

MALONE, T.W., GRANT, K.R. and TURBAK, F.A. (1986) The information lens: an intelligent system for information sharing in organization. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17, 1986), ACM, New York, 1-8.

MARK, W. (1986) Knowledge-Based Interface Design. In a D.A. NORMAN and S.W. DRAPER (eds.) User Centered System Design, New Perspectives on Human-Computer Interaction, Hillsdale, New Jersey: Lawrence Earlbaum Associates Publishers.

MARTIN, J. (1973a) Design of man-computer dialogues. Englewood Cliffs, N.J. Prentice-Hall.

MARTIN, T.H. (1973b) The User Interface in Interactive Systems, Annual Review of Information Science and Technology, 8, 203-219.

MAYER, R.E. (1975) Different Problem Solving Competencies established in Learning Computer Programming with and without Meaningful Models. Journal of Educational Psychology, 67, 725-734.

MAYER, R.E. (1976) Some Conditions of Meaningful Learning for Computer Programming: Advance Organisers and Subject Control of Frame Order. Journal of Educational Psychology, 47, (6), 143-150.

McCLELLAND, J.L., RUMELHART, D.E. and the PDP Research Group, (1986) Parallel Distributed Processing, Vol. II, Cambridge, MA.: MIT Press.

McCLOSKEY, M. (1983) Naive theorie of motion. In D. GENTNER, and A.L. STEVENS (eds.), Mental Models, Hillsdale, New Jersey: Lawrence Earlbaum Associates.

Mc CORMICK, E.J. and SANDERS, M.S, (1982) Human Factors in Engineering and Design. New York: McGraw-Hill.

McDONALD, N.H. (1982) A multi media approach to the user interface. In Y. VASSILIOU (ed.) Human Factors and Interactive Computer System. Proc. of the New York Symposium on User Interfaces, (New York, May 26 to 28, 1982), Norwood, New Jersey: Ablex Publishing Corporation. 47-82.

McKENDREE, J. and CARROLL, J.M. (1986) Advising Roles of a Computer Consultant. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, 35-40.

MELTON, A.W., and BRIGGS, G.E. (1960) Engineering psychology. Annual review of psychology, 11, 71-98.

MINSKY, M.L. (1975) A Framework for Representing Knowledge. In P. WINSTON (ed.), The Psychology of Computer Vision. New York; McGraw-Hill.

MIYATA, Y. and NORMAN, D.A. (1986) Psychological issues in support of multiple activities. In D.A. NORMAN, and S.W. DRAPER (eds.) User centered system design. Hillsdale, New Jersey: Lawrence Earlbaum Associates.

MORAN, T.P. (1978) Introduction to the Command Language Grammar. Xerox Palo Alto Research Center, Palo Alto, CA.

MORAN, T.P. (1981a) An Applied Psychology of the User. Computing Surveys, 13, 1, 1-11.

MORAN, T.P. (1981b) The Command Language Grammar: a representation for the user interface of interactive computer systems. International Journal of Man-Machine Studies, 15, 3-50.

MOSES, F.L. and EHRENREICH, S.L. (1981) Abbreviations for automated systems. Paper presented at the Human Factors Society Meeting, Rochester, New York.

MOSIER, J.N. and SMITH, S.L. (1986) Application of Guidelines for Designing User Interface Software. Behaviour and Information Technology, 5 (1), 39-46.

NAISBITT, J. (1984) Megatrends. New York: Warner Brother.

NAKATANI, L.H., EGAN, D., RUEDISUELI, L.W., HAWLEY, P.M. and LEWART, D.K. (1986) TNT: A Talking Tutor "N" Trainer for Teaching the Use of Interactive Computer Systems. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, 29-34.

NEWELL, A. (1973) You can't play twenty questions with nature and win. In W.G. CHASE (ed.), Visual Information Processing. New York: Academic Press.

NEWELL, A. and CARD, S.T. (1985) The prospect for psychological science in human-computer interaction. Human Computer Interaction, 1, 209-242.

NEWELL, A. and SIMON, H.A. (1972) Human Problem Solving. Englewood Cliffs, New Jersey: Prentice-Hall.

NICKERSON, R.S. (1977). On Conversational Interaction with Computers. In S. TREU (ed.) User-Oriented Design of Interactive Graphic Systems, Association for Computing Machinery, New York, 101-103.

NICKERSON, R.S. (1981) Why interactive computer systems are sometimes not used by people who might benefit from them. International Journal of Man-Machine Studies, 15, 469-483.

NICKERSON, R.S. (1986) Using Computers: Human Factors in Information Systems. Cambridge, Mass.: Bradford Book, MIT Press.

NORMAN, D.A. (1980) Cognitive Engineering and Education. In D.T. TUMA and F. REIF (eds.), Problem-Solving in Education, Hillsdale, New Jersey: Lawrence Earlbaum Associates, 97-107.

NORMAN, D.A. (1981) The Trouble with Unix. Datamation, 27, 139-150.

NORMAN, D.A. (1983a) Design Principles for Human-Computer Interfaces. Proc. CHI'83, Human Factors in Computing Systems, (Boston, December 12 to 15), ACM, New York, 1-10.

NORMAN, D.A. (1983b) Some observations on mental models. In D. GENTNER, and A.L. STEVENS (eds.), Mental Models, Hillsdale, New Jersey: Lawrence Earlbaum Associates.

NORMAN, D.A. (1984) Stages and Levels in Human-Machine Interaction. International Journal of Man-Machines Studies, 21, 365-375.

NORMAN, D.A. (1986) Cognitive Engineering. In Norman and Draper (eds.): User Centered System Design, New Perspectives on Human-Computer Interaction. Hillsdale, New Jersey: Lawrence Earlbaum Associates.

NORMAN, D.A. and DRAPER, S.W. (1986) User Centered System Design, New Perspectives on Human-Computer Interaction. Hillsdale, New Jersey: Lawrence Earlbaum Associates.

OBORNE, D.J. (1982). Ergonomics at work. Chichester: John Wiley and Sons.

OGDEN, W.C. and BROOKS, S.R. (1983) Query Languages for the casual user: exploring the middle ground between formal and natural languages. Proc. CHI'83, Human Factors in Computing Systems, (Boston, December 12 to 15), ACM, New York, 161-165.

OREN, S.S. (1974) A Mathematical Theory of Man-Machine Text Editing. IEEE Transaction on Systems, Man, Cybernetics, SMC-14 (3), 258-267.

PASK, G. (1976) Styles and strategies of learning. British Journal of Educational Psychology, 46, 128-148.

PAXTON, A.L. and TURNER, E.J. (1984) The application of human factors to the needs of the novice computer user. International Journal of Man-Machine Studies, 20, 137-156.

PAYNE, S.J. (1985) Task-Action Grammars: the Mental Representation of Task Languages in Human-Computer Interaction. Doctoral thesis, Sheffield University.

POLLER, M.F. and GARTER, S.K. (1983) A Comparative Study of Moded and Modeless Text Editing By Experienced Editor Users. Proc. CHI'83 Human Factors in Computing Systems, (Boston, December 12 to 15, 1983), ACM, New York, 166-170.

POLSON, P.G. and KIERAS, D.E., (1985) A Quantitative Model of the Learning and Performance of Text Editing Knowledge. Proc. CHI'85 Human Factors in Computing Systems, (San Francisco, April 14 to 18, 1985), ACM, New York, 207-212.

POLSON, P.G., MUNCHER, E. and ENGELBECK, G. (1986) A Test of a Common Elements Theory of Transfer. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17, 1986), ACM, New York, 78-83.

QUINN, L. and RUSSELL, D.M. (1986) Intelligent Interfaces: User Modes and Planners. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, 314-320.

RAMSEY, H.R. and ATWOOD, M.E. (1979) Human Factors in Computer Systems: A review of the literature. Technical Report #SAI-79-111-DEN, Englewood, Co.: Science Applications, Inc.

RAMSEY, H.R. and GRIMES, J.D. (1983) Human Factors in Interactive Computer Dialog. Annual Review of Information Sciences and Technology, 18, 29-59.

REICHMAN, R. (1985) Getting computers to talk like you and me. Cambridge, Mass.: MIT Press.

REISNER, P. (1981) Human factor studies of database query languages: a survey and assessment. Computing Survey, 3, (1).

RICH, E. (1983) Users are individuals: individualizing user models. International Journal of Man-Machine Studies, 18, 199-214.

RISSLAND, E.L. (1984) Ingredients of Intelligent User Interfaces. International Journal of Man-Machines Studies, 21, 377-388.

ROBERT, J-M. (1986a) Some Highlights of Learning by Exploration. Proc. International Scientific Conference on "Work with Display Units", (Stockholm, May 12 to 15), Sweden, 348-353.

ROBERT, J.-M. (1986b) An Approach for Verifying the Consistency of User Interfaces. Proc. International Scientific Conference on "Work with Display Units," (Stockholm, May 12 to 15), Sweden, 516-521.

ROBERTS, T.L. (1979) Evaluation of Computer Text Editor, Doctoral thesis, Computer Science Department, Standford University. Reprinted as Technical report #SSL79-9, Xerox Palo Alto Research Center, Palo Alto, CA.

ROBERTS, T.L. and MORAN, T.P. (1982) Evaluation of Text Editors, Proc. CHI'82, Human Factors in Computing Systems. (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 136-141.

ROBERTS, T.L. and MORAN, T.P. (1983) The Evaluation of Text Editors: Methodology and Empirical Results. Communication of ACM, 26, (4), 265-283.

ROBERTSON, I.T. (1985) Human information-processing strategies and style. Behaviour and Information Technology, 14, 1, 19-29.

ROBERTSON, S.P. and BLACK, J.B. (1983) Planning Units in Text Editing Behavior. Proc. CHI'83 Human Factors in Computing Systems, (Boston, December 12 to 15, 1983), ACM, New York, 217-221.

ROBERTSON, G., McCRACKEN, D. and NEWELL, A. (1981) The zog approach to man-machine communication. International Journal of Man-Machine Studies, 14, 461-488.

ROEBUCK, J.H., KROEMER, K.H. and THOMPSON, W.G. (1975) Engineering Anthropometry Methods. New York: John Wiley and Sons.

ROEMER, J.M. and CHAPANIS, A. (1982) Learning performance and attitudes as a function of the reading grade level of a computer-presented tutorial. Proc. CHI'82 Human Factor in Computing Systems, (Gaitherburg, Maryland, March 15 to 17), ACM, New York, 239-244.

ROHR, G. (1986) Conceptual primitives and their representation in understanding complex operations. Proceedings of the 3rd European Conference on Cognitive Ergonomics, (Paris, September 15 to 19, 1986), INRIA, 165-168.

ROSENBERG, J.A. (1982) Evaluating the suggestiveness of command names. Behaviour and information technology, 1, 371-400.

ROSENBERG, J.A. (1983) Featural Approach to Command Names. Proc. CHI'83, Human Factors in Computing Systems, (Boston, December 12 to 15), ACM, New York, 116-119.

ROSS, B.H. and MORAN, T.P. (1983) Remindings and their Effects in learning a text editor. Proc. CHI'83, Human Factors in Computing Systems, (Boston, December 12 to 15, 1983), ACM, New York, 222-225.

ROSSON, M.B. (1983) Patterns of experience in text editing. Proc. CHI'83, Human Factors in Computing Systems, (Boston, December 12 to 15), ACM, New York, 171-175.

RUMELHART, D.E. (1975) Notes on a Schema for Stories. In D.G. BOBROW, and A.M. COLLINS, (eds.), Representations and Understanding: Studies in Cognitive Science, New York: Academic Press.

RUMELHART, D.E., McCLELLAND, J.L. and the PDP Research Group (1986) Parallel Distributed Processing, Vol. I, Cambridge, MA.: MIT Press.

RUMELHART, D.E. and NORMAN, D.A. (1982) Simulating a skilled typist: A study of skilled cognitive-motor performance. Cognitive Science, 6, 1-36.

SACERDOTI, E.D. (1977) A structure for plans and behavior. New York: Elsevier.

SANFORD, D.L. (1986) Using metacommunication to generate human-human communication protocols at the human-computer interface. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13-17), ACM, New York, p. 353.

SAUTER, S.L., GOTTLIEB, M.S., JONES, K.C., DODSON, B.N. and ROHRER, K.M., (1983) Job and health implications of VDT use: Initial results of the Wisconsin-NIOSH study. Communication of the ACM, 26, (4), 264-294.

SCAPIN, D. (1981) Computer commands in restricted natural language: Some aspects of memory and experience. Human Factors, 23, 365-375.

SCAPIN, D. (1982) Computer Commands Labelled by Users Versus Imposed Commands and the Effect of Structuring Rules on Recall. Proc. CHI'82, Human Factors in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 17-19.

SCHANK, R.C. (1972) Conceptual Dependency: A Theory of Natural Language Understanding. Cognitive Psychology, 3, 552-631.

SCHANK, R.C. and ABELSON, R. (1977) Scripts, Plans, Goals and Understanding. Hillsdale, New Jersey: Lawrence Earlbaum Associates.

SCHMANDT, C. (1985) Voice Communication with Computers. In H.R. HARTSON (ed.) Advances in Human-Computer Interaction, Vol. I, Norwood, New Jersey: Ablex Publishing Corporation.

SCHMANDT, C. and HULTEEN, E.A. (1982) The Intelligent Voice-Interactive Interface. Proc. CHI'82, Human Factors in Computing Systems, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 363-366.

SCHNEIDER, M.L., NUDELMAN, S. and HIRSH-PASEK, K. (1982) An analysis of line numbering strategies in text editors. Proc. CHI'82 Human Factor in Computing System, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 148-151.

SCIENCE COUNCIL OF CANADA. (1982) Planning now for an information society. Hull: Minister of Supply and Services Canada.

SCIENCE COUNCIL OF CANADA. CORDELL, A.J. (1985) The Uneasy Eighties. The Transition to an Information Society. (Background Study; 53) Hull: Minister of Supply and Services Canada.

SEARLE, J.R. (1969) Speech Acts. Cambridge: Cambridge University Press.

SEIBEL, R. (1972) Data entry devices and procedures. In H.P. VAN COTT and R.G. KINKADE (eds.) Human Engineering Guide to Equipment Design. Revised ed. Washington, D.C.: U.S. Government Printing Office. 311-344.

SHACKEL, B. (1984) Information Technology - A challenge to ergonomics and design. Behaviour and Information Technology, 3, (4) 263-275.

SHACKEL, B. (1985) Ergonomics in Information Technology in Europe - A Review. Bahaviour and Information Technology, 4, (4) 263-287.

SHNEIDERMAN, B. (1979) Human Factors Experiments in Designing Interactive Systems. Computer, 12, 9-24.

SHNEIDERMAN, B. (1980a) Software Psychology, Cambridge, MA.: Winthrop.

SHNEIDERMAN, B. (1980b) System Message design: Guidelines and experimental results. In A. BADRE and B. SHNEIDERMAN (eds.), Directions in Human-Computer Interaction, Norwood, New Jersey: Ablex Publishers.

SHNEIDERMAN, B. (1982) The future of interactive systems and the emergence of direct manipulation. Behaviour and Information Technology. 1, (3), 237-256.

SHNEIDERMAN, B. (1986a) Direct manipulation: an object-oriented visual programming language. Tutorial: CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York.

SHNEIDERMAN, B. (1986b) Seven plus or minus two central issues in human-computer interaction. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17), ACM, New York, 343-349.

SHURTLEFF, D.A. (1980) How to make displays legible. La Mirada, CA.; Human Interface Design.

SLEEMAN,D. and BROWN, J.S. (1982) Intelligent Tutoring Systems, New York: Academic Press.

SMALL, D.W. (1983) An experimental comparison of natural and structured Query languages. Human Factors, 25, (3), 253-263.

SMITH, S.L. (1986) Standards versus Guidelines for Designing User Interface Software. Behaviour and Information Technology, 5, (3), 47-61.

SMITH, S.L. and MOSIER, J.N. (1984) The User Interface to Computer-Based Information Systems: a Survey of current Software Design Practice. Behaviour and Information Technology, 3, (3), 95-203.

STEVENS, G.C. (1983) User-friendly Computer Systems? A critical Examination of the Concept. Behaviour and Information Technology, 2, (1), 3-16.

STEWART, T.F.M. (1981) The Specialist User. In B. SHACKEL (ed.) Man- Computer Interaction: Human factors aspects of computers and people. Rockville, Maryland: Sijthoff and Noordhoff.

STREITZ, N.A. (1986) Cognitive Ergonomics: An approach for the design of user-oriented interactive systems. In F. KLIX, and H. WANDKE, (eds.), Proc. of the First Network Seminar of the International Union of Psychological Science on Man Computer Interaction Research; MACINTER - I (Berlin RDA, October 16 to 19), Amsterdam: North-Holland, 21-34.

SULLIVAN, M. and COHEN, P.R. (1985) An Endorsement-Based Plan Recognition Program. Proc. Ninth International Joint Conference on Artificial Intelligence, AIJCHI'85, (Los Angeles, August 18 to 23) CA., Vol.1, 475-479.

TAGG, W. (1974) Programming languages for school use. Computer Education, 16, 11-22.

TEITELBAUM, R.C. and GRANDA, R.E. (1983) The Effects of Positional Constancy on Searching Menus for Information. Proc. CHI'83, Human Factors in Computing Systems, (Boston, December 12 to 15), ACM, New York, 150-153.

THOMAS, J.C. and SCHNEIDER, M.L. (1984), Human Factors in Computer Systems. Norwood, New Jersey: Ablex Publishing Corporation.

TOFFLER, A. (1980) The Third Wave. New York: Bantam.

TOONG, H.D. and GUPTA, A. (1982) Personal Computer. Scientific American, 247 (6), 86-107.

TOU, F.N., WILLIAMS, M.D., FIKES, R., HENDERSON, R. and MALONE, A.T. (1982) RABBIT: An Intelligent Database Assistant. Proc. of the AAAI'82 Cognitive and Instructional Sciences Group, (Pittsburg, August 16 to 20), Xerox PARC, CA.

TURING, A.M. (1950) Computing Machinery and Intelligence. Mind, 59, 433-450.

TURNER, J.A., JARKE, M., STOHR, E.A. and VASSILIOU, Y. (1984) Using restricted natural language for data retrieval: a plan for field evaluation. In Y. VASSILIOU (ed.) Human Factor and Interactive Computer System. *Proc. of the New York Symposium on User Interfaces*, (New York, May 26 to 28, 1982), Norwood, New Jersey: Ablex Publishing Corporation. 163-190.

TUROFF, M., HILTZ, S.R. and KERR, E.B. (1982) Controversies in the design of Computer-Mediated Communications Systems: A Delphi Study. *Proc. CHI'82 Human Factor in Computing Systems*, (Gaithersburg, Maryland, March 15 to 17), ACM, New York, 89-100.

TYLER, S., ROTH, S. and POST, T. (1982) The Acquisition of Text Editing Skills. *Proc. CHI'82 Human Factors in Computing Systems*, (Gaithersburg, Maryland, March 15 to 17, 1982), ACM, New York, 324-325.

VALLEE, J. (1976) There ain't no user science: A tongue-in-cheek discussion of interactive systems. *Proc. of the American Society for Information Science Annual Meeting* 13, San Francisco.

VAN DER VEER, G.C. and BEISHUIZEN, J.J. (1986) Learning styles in conversation - a practical application of Pask's learning theory to human-computer interaction. In F. KLIX, and H. WANDKE (eds.) *Proc. of the First Network Seminar of the International Union of Psychological Science on Man Computer Interaction Research: MACINTER - I*, (Berlin RDA, October 16 to 19), Amsterdam: North-Holland, 195-205.

VAN DER VEER, G.C., TAUBER, M.J., WAERN, Y. and VAN MUYLIJK, B. (1985) On the interaction between system and user characteristics. *Behaviour and Information Technology*, *4*, 4, 289-308.

VASSILIOU, Y. (1984) Human Factors and Interactive Computer System. *Proc. of the New York Symposium on User Interfaces*, (New York, May 26 to 28, 1982), Norwood, New Jersey: Ablex Publishing Corporation.

VASSILIOU, Y. and JARKE, M. (1984) Query Language - A taxonomy. In Y. VASSILIOU (ed.) Human Factors and Interactive Computer System. *Proc. of the New York Symposium on user Interfaces*, (New York, May 26 to 28, 1982), Norwood, New Jersey: Ablex Publishing Corporation. 47-82.

WEBSTER, W.G., WALKER, A.D. (1981) Problem solving strategies and manifest brain assymetry. *Cortex*, *14*, 474-484.

WEIZENBAUM, J. (1976) *Computer power and human reason: From judgment to calculation*. San Francisco: Freeman.

WHITESIDE, J., JONES, S., LEVY, P.S. and WIXON, D. (1985) User Performance with Command, Menu and Iconic Interfaces. *Proc. CHI'85 Human Factors in Computing Systems*, (San Francisco, April 14 to 18), ACM, New York, 185-191.

WINOGRAD, T., DAVIS, R. and DREYFUS, S. (1985) Expert systems: How far can they go? Proc. Ninth International Joint Conference on Artificial Intelligence, AIJCHI'85, (Los Angeles, August 18 to 23) CA., Vol 2, p. 1305.

WINOGRAD, T. and FLORES, F. (1986) Undertanding Computers and Cognition; a new Foundation for Design. Norwood, New Jersey: Ablex Publishing Corporation.

WISER, M. and CAREY, S. (1983) When heat and temperature were one. In D. GENTNER, and A.L. STEVENS (eds.), Mental Models, Hillsdale, New Jersey: Lawrence Earlbaum Associates.

WRIGHT, P. and LICKORISH, A. (1983) Proof-reading texts on screen and paper. Behaviour and Information Technology, 2, (3), 227-235.

YAMADA, H. (1980) A historical study of typewriting and typing methods: From the position of planning japanese parallels. Journal of Information Processing, 2, 179-202.

YOUNG, R.M. (1981) The Machine Inside the Machine: Users' Models of Pocket Users' Calculators. International Journal of Man-Machine Studies, 15, 51-85.

YOUNG, R.M. (1983) Surrogates and mappings: two kinds of conceptual models for interactive devices. In D. GENTNER, and A.L. STEVENS (eds.), Mental Models, Hillsdale, New Jersey: Lawrence Earlbaum Associates.

ZIEGLER, J.E., HOPPE, H.U. and FAHNRICH, K.P. (1986) Learning and Transfer for Text and Graphics Editing With a Direct Manipulation Interface. Proc. CHI'86 Human Factors in Computing Systems, (Boston, April 13 to 17, 1986), ACM, New Yrok, 72-77.

ZEIGLER, J.E., VOSSON, P.H. and HOPPE, H.U. (1986) On using production systems for cognitive task analysis and prediction fo transfer of cognitive skill. Actes de la 3e conférence européenne d'ergonomie cognitive, (Paris, September 15-19, 1986), INRIA, 115-128.

ZLOOF, M.M. (1983) The Query-by-Example concept for user-oriented business systems. In M.E. SIME, and M.J. COOMBS, (eds.): Designing for Human-Computer Interaction. London: Academic press. 285-310.